

nm

nascom microcomputers limited

NAS-DIS

disassembler

for NASCOM

microcomputers

PART NO.

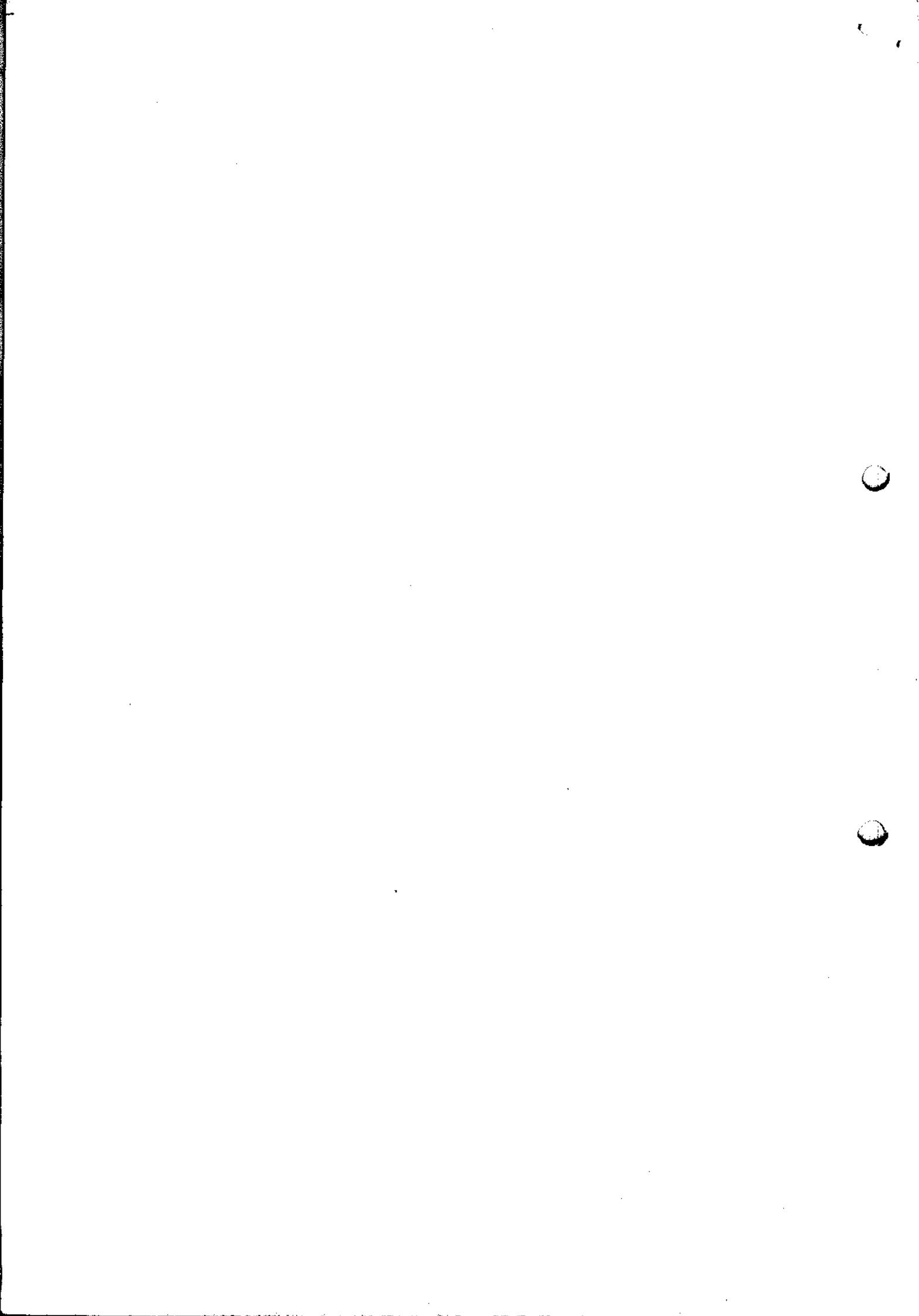
DATE

ISSUE

230-300

21-04-80

2



INTERACTIVE OPTION SUMMARY SHEET

- Source form only
- Tape output (or printer)
- ZEAP file to memory
- Xreference listing
- Labels
- Paginated listing
- Delay at end-of-line
- Range limitation
- Unmodified RST handling

Options for source file

- ZEAP file:- Z (RU)
- Tape file:- STLD (RU)

Full listing to printer

- TLP(X) (RU)

Note: All address pairs entered are inclusive. ie Start End.

C

C

CONTENTS

	page
1. INTRODUCTION	2
2. MEMORY REQUIREMENTS	3
3. RUNNING THE DIS-ASSEMBLER	4
3.1 Simple mode	4
3.2 Interactive mode	5
4. NOTES ON USING THE DIS-ASSEMBLER	9
5. PRODUCING SOURCE FILES	10
5.1 ZEAP files direct to memory	10
6. USING A PRINTER	11

APPENDICES

A IDIOSYNCRASIES AND BUGS	12
B MODIFICATIONS	13
C WRITING YOUR OWN PROGRAM FOR REVAS	14
D SAMPLE OUTPUT - SIMPLE MODE	15
E SAMPLE OUTPUT - INTERACTIVE MODE	16
F PROGRAM LISTINGS	18

1. INTRODUCTION

An assembler takes in instructions such as LD (HL),A INC HL CPIR and converts them to the corresponding machine code (eg 77 23 ED B1). A dis-assembler, or reverse assembler, performs the complementary function. It translates machine code back to mnemonics.

The dis-assembler is a useful tool. It lets you examine programs in memory in an easily readable format. You can produce "listings" of programs for which you do not have the source, and, working from them, you can make changes to programs to tailor them more to your own personal requirements. If the changes you wish to make are extensive, you can generate a source file for the program, which can be edited and re-assembled, perhaps with new origins.

The dis-assembler can be run in two modes, a simple one for direct examination of memory, and a more complex one for producing labelled listings and source files. You will probably find yourself using the simple mode often as it provides a very useful extension to the monitor in that it produces easily readable tabulations of programs in memory. The full interactive mode is of use when you buy a new piece of software for which you wish to produce a source listing/file, and, as a result, is likely to be used less frequently.

One thing must be realised from the outset. An original assembly language program contains a lot of information that is only of use to the programmer. These are the comments and label names which are used to document the program and to make it readable. The dis-assembler has no way of re-inserting these as they have no representation in the machine code, but it can generate a form of label and add a little more information to its output which is of help in following a listing. (When you come to examine a large dis-assembled listing you will realise the value of plentiful comments and informative labels!).

The format of the output produced by the dis-assembler is:-

AAAA OP OP OP OP LLLL MMMM OPERAND ;CCCC

Where:AAAA is the address of the instruction.

OP are the bytes (1 to 4) making up the instruction

LLL is a label (if present).

MMMM is a 2 to 4 character mnemonic.

OPERAND is the operand field. (eg A,(HL) or HL,f1025)

CCCC are the 7-bit ASCII characters corresponding to bytes 1 to 4 of the instruction. All control characters appear as a period.

The kernel of the package is the reverse assembler subroutine REVAS, and if you wish to write your own programs to drive it the necessary information is given in appendix C. (Don't just sit there thinkingif only it..... : Sit down and write your own control program to do what you want).

REVAS has been written to produce ZEAP compatible mnemonics, and in addition, recognises three special instructions. These are the three Restart codes D7,DF,E9 which, if they occur in a program that runs with Nas-Sys, are followed by one or more parameters. (The routines are RCAL, SCAL, and PRS). REVAS lists the parameters directly after printing the restart instruction, and does not attempt to interpret them as program.

2. MEMORY REQUIREMENTS

NAS-DIS is supplied in three 2708s and is coded to run at address C400. When used with the NASCOM2 it is expected that these will be inserted into the spare sockets on the main processor board. Refer to the N2 hardware manual and check that you have the header plug wired correctly, the strapping fields (LKB2-4) correctly set, and LSW1 7&8 suitably set.

Assuming that you have the eight RAM/EPROM sockets decoded for address range C000-DFFF insert the three EPROMs as follows, making sure that you have them correctly orientated:-

NASDIS-1 in socket IC36
NASDIS-2 in socket IC37
NASDIS-3 in socket IC38

Remember that EPROMs are MOS devices and can be damaged by static electricity, so take reasonable care while you are handling them.

When running in the full interactive mode the program uses workspace from E00 to EC2. When only run in the simple mode the workspace used is E00 to E43. In addition some memory is required for the dis-assembler's stack. NAS-DIS does not set the stack pointer at all, it just accepts the value passed to it on entry. Conventionally the program is run by the Nas-sys "E" command, and in this case the stack area used is from FFF downwards (towards 0).

In the simple mode the stack can reach FEO, while in the interactive mode it can reach FC8. However in the interactive mode any address pairs entered in response to the prompt "Data Areas?" are saved on the stack, so the stack depth will increase by four bytes for every pair entered. This means that there is room for about 60 address pairs before the dis-assembler is likely to overwrite its own workspace, and about 36 address pairs before it overwrites ZEAP's workspace (see section 5.1).

C000
C400
DFFF

3. RUNNING THE DIS-ASSEMBLER

The dis-assembler has two modes of operation, a simple one for direct examination of areas of memory, and a more complex one for producing labelled listings and/or assembler files.

REVAS is run by using the monitor E command. Three possible formats are acceptable:

```
E C400  
E C400 XXXX  
E C400 XXXX YY
```

In the first instance, where no parameters are passed to the program, the dis-assembler runs in its interactive mode and will prompt you for further information (see on). Where one or two extra parameters are included in the command line the simple mode is executed.

3.1 Simple mode

Where both parameters are included in the command line the dis-assembler will dis-assemble code starting at address XXXX and will pause after listing YY lines in mnemonic-type format. If the second extra parameter is omitted then a default value of 1 line is assumed.

Depressing any key on the keyboard will produce the next YY lines, return to the monitor only occurring when "Escape" (shift/enter) is typed.

This mode provides an easy way of displaying areas of memory in a mnemonic format and, amongst other uses, provides an admirable way of checking that any Hex codes you have entered manually to memory are actually the instructions you intended.

In this mode the dis-assembler assumes that the three RST instructions D7,DF and EF are RCAL,SCAL and PRS.

3.2 INTERACTIVE MODE

You will be prompted first to enter the options to be used for the dis-assembly as shown below:

Options (STZXLDRU) -

You select options by entering the corresponding letters followed by a carriage return. If you require more than one option, enter them all before the carriage return. Any unrecognised character will cause the question to be repeated. In this case all the options previously entered will be cleared.

The option letters have the following significance:

OPTION X (Cross-reference)

A listing of all referenced addresses and the corresponding addresses of the instructions that refer to them will be produced at the end of the dis-assembly. See the sample output for details.

OPTION L (Labelled)

The dis-assembler outputs a label symbol to the left of any instruction which is referred to elsewhere in the dis-assembled program. The form of the label is an L followed by the address of the label. See the sample output.

OPTION T (Tape/typed output)

Selecting this option causes the screen output to be echoed to the serial output port where it could be printed, or it could be recorded on tape for re-input later (say to an assembler). See also options D,L and S which may be relevant if option T is selected.

OPTION S (Source form only)

This option is useful if you want to re-input the data as source for an assembler, when only the text and label references are required. It causes the listing of the location of each instruction and the code bytes of each instruction to be omitted. Also all non-significant blanks are removed from the output and, except in the case of DEFB lines, the comment field is deleted. See also options D and T which may be relevant.

OPTION D (Delay)

This option introduces a short delay after each line is printed. It is used for printers that require a small delay at the end of a line, and also when producing a source file to tape (see on).

OPTION Z (ZEAP source file)

This option causes a ZEAP compatible source file to be produced directly in memory complete with all necessary line numbers and pointers. This is covered in more detail in section 5.

OPTION P (Paginated output)

This option formats the output into pages with a title at the top of each page and a page number.

OPTION R (Range limitation)

This option allows you to restrict the range of the dis-assembly listing whilst actually dis-assembling a larger area of memory. (If you wish to modify a small part of a large program this allows you to list only the area of interest whilst ensuring that any possible entry points in it are labelled).

OPTION U (Unmodified RST handling)

This option cancels the special handling of the three RST instructions - RCAL, SCAL and PRS. It would be used if you are dis-assembling a program that was written to run with a monitor that used the RST instructions in a different manner to Nas-sys.

If you selected options L or X you will now see the prompt:

Symbol table area?

The dis-assembler now wants the start and finish address of some memory that it can use as temporary workspace to hold the symbol table that it has to generate. Efficient use is made of the memory allocated, but if insufficient memory is specified an error message will appear during Pass 1 (see on).

Enter the two addresses followed by a carriage return.

If you selected option Z you will see the prompt:

ZEAP file area?

This prompt appears in place of the one above. The program uses the top of this area to hold its symbol table, and stores the ZEAP source file at the bottom. The addresses entered here are also passed on to ZEAP. In this instance more memory is required to hold the file in addition to the symbol table, and so overflow may also occur during pass 2, while the source file is actually being stored in the file area (see section 5.1).

If you selected option P you will now see the prompt:

Title?

Enter the title you wish to appear on the listing. At this time you also have the opportunity to redefine the lines/page and lines between pages figures used by the program. If you wish to do this you should start the title line:-

=AABB T.....T

Where AA is the hexadecimal number of lines between pages+1
and BB is the hexadecimal number of lines/page
and T.....T is the title.

The total number of lines per page are AA+BB+1.

You will now see the prompt:

What on?

The program is now asking you for the start and finish address of the area of memory you wish to dis-assemble. You also have the option of entering a third address as well. If you only enter two addresses REVAS will do a straightforward dis-assembly of the area specified. However if the third address is entered, the program will still dis-assemble the same area of memory, but the addresses on the listing will start with the third address. ie The area of memory will be dis-assembled as though it were actually located at the third address entered, rather than the first.

This allows you to correctly dis-assemble any program, even if it is not at its execution address. (That area of memory may already be in use in your system, or may even be absent!).

Enter the two (or three) addresses separated by spaces and finish with a carriage return.

Any addresses you enter from now on refer to the listing address. This address will differ from the memory address if you responded with three parameters to the last request.

If you selected option R you will now see the prompt:

Listing range?

Enter the start and finish address of the desired range.

NOTE If the addresses do not overlap the full dis-assembly range then nothing will be listed except the cross-reference table (assuming the X option has also been selected).

Data areas?

DATA AREAS in this context refers to areas of data within the bounds of the START and FINISH addresses which you do not want to be dis-assembled into instruction format. Typically they would be data tables, messages etc. If you know of such areas in the program enter a pair of start and finish addresses for each area, one pair per line. The address pairs should be entered in ascending order. You will get an error message if the pair of addresses are unacceptable and the cursor will be positioned to the start of the line in error. Enter a carriage return on its own to terminate the data area list. If you notice that you have entered an incorrect address in a previous entry typing "--" (followed by enter) will delete the last entry. This can be used repeatedly until you reach the line in error.

NOTE. The parameters following the restart calls RCAL/SCAL/PRS are dealt with automatically by REVAS and they should not be identified here, although remember that this may have been cancelled by option U.

Finally the program prompts:

Go?

The program is waiting for you to press a key on the keyboard before it runs. If you press the spacebar the program will start to run in a single step mode, any other key resets the single step flag.

If you have selected options L, Z or X the blinking cursor will disappear and there will be a delay of a few seconds or more while the dis-assembler makes a first pass of the code building up its symbol table. In the event of it running out of memory for the symbol table the message "Overflow" will appear on the screen along with the line currently being dis-assembled displayed below it. This shows you how far the program has got and should let you make a reasonable estimate of how much more memory is required for the symbol table. In the event of overflow the program has to be re-run from the beginning.

At the start of Pass 2, (immediately if neither option L, Z or X were entered), the listing will start to appear on the screen. While the program is listing you can switch easily between a continuous and a single step mode. During a continuous listing pressing any key on the keyboard halts it at the end of the line currently being listed. (The keyboard is only checked at the end of the line, so at low listing speeds make sure you hold the key down until the end of the line is reached). Pressing another key allows the listing to resume. If the 'resume' key is the spacebar the program will automatically halt after listing the next line. With any other key it will continue to list until you interrupt it again by pressing a key on the keyboard. Typing "Escape" (shift/enter) at this point will abort the program and return control to the monitor.

4. NOTES ON USING THE DIS-ASSEMBLER

1. Any program, including the operating system, REVAS itself, ZEAP etc, can be dis-assembled if it is in memory with the dis-assembler. Be careful that the program to be dis-assembled does not overlap the workspace used by REVAS. If it does, move it to a different area of memory and respond with three addresses to the prompt - What on?
2. Unless you specify data areas, REVAS has no way of knowing whether the bytes it is trying to decode are program or data. It assumes that they are program, unless instructed otherwise, so if a program contains unspecified embedded data, then this data will be decoded as program.

Due to the structure of the Z80 instruction set, it is unlikely that invalid instructions will be detected in such areas, but they are easily recognisable as the sequence of instructions will make no sense at all. In the case of an ASCII message the text will be plainly visible in the comment field on the right of the listing.

At the end of the data, it may decode a few instructions incorrectly. This is because it may be getting the boundary between individual instructions wrong. A similar effect occurs if you ask it to start half way through an instruction.

3. LABELS: There is no easy way for the dis-assembler to decide whether the operand in an instruction such as LD HL, £1000 is an address, byte count, or arithmetic constant, and so it assumes that it is an address and generates a label for it. As a result any source files produced by the dis-assembler will probably require a little editing before they can be re-assembled to run at another address.

NOTE: Any labels that lie outside the bounds of the dis-assembled code are equated to their absolute values (see appendix E). If a label lies within the bounds specified, but does not line up with an instruction boundary, then a program relocatable EQU statement is generated.

eg L147A EQU \$-2

4. For those who wish to examine a full dis-assembly listing, but do not possess a printer, the suggested procedure is to generate the listing to tape. This tape can then be played back through the serial port and sections of the listing examined on the screen. It is a simple matter to stop/start the tape recorder, and to fast wind forward and backward to locate any place in the listing.

5. The delay given by option D should be adequate for most cases, but if it is found to be too short it can be effectively doubled by halving the CPU clock rate before running the dis-assembler (and doubling it afterwards).

5. PRODUCING SOURCE FILES

The following is a suggested procedure for generating source files.

First of all, using the simple mode, quickly run through the program to be dis-assembled and note down all the obvious data areas within the program. (This will not be necessary if you have already produced a source listing of the program as you can work from that).

Next run the dis-assembler in its interactive mode and, if your assembler is ZEAP (version 2.0) and you have sufficient room in memory for the source file and the symbol table, use only option Z. Otherwise use options LSTD (Labels,Source,Tape,Delay), and record the file on tape. Answer the remaining prompts and run the program.

If you have created the file on tape you will have to read it back into your assembler/editor. If (like ZEAP) it has an "auto line number" mode, or accepts lines directly, then enter your editor, set Auto mode if necessary, and play the source tape back through the tape interface. The delay at the end of the line gives the editor time to move the line into the file and generate the next line number before the next line starts. If your editor will not accept the tape directly, you will have to write a small program of your own to read the tape in. (Either load it directly to memory in the correct format, or if the problem is a minor one write your own driver to pre-process the incoming line - see Nas-sys U command).

Having got the source file into your editor the first thing to do is to set an ORG statement. Then, no doubt, there will be various areas you will want to edit - Data areas may need changing to DEFMs and DEFWs (strings and address tables) and some extra labels added as a result. Before you make any change to the source code which will result in different object code (such as a new ORG or a small modification), it is advisable to assemble the file and, using a simple program, check that the object code produced is identical to the original. This confirms that you have made no errors in your editing so far (like leaving a significant trailing space out of a DEFM) and could save you a lot of time later.

5.1 ZEAP files direct to memory

By using option Z a ZEAP file can be created directly in memory without passing through the intermediate stage of a tape file. Before this option is used ZEAP should have been entered and exited to ensure that it has initialised its own workspace. The actual file area to be used is set by the addresses entered in response to the prompt from the dis-assembler.

During Pass 1 the dis-assembler constructs the symbol table at the top of this area. In the event of it having insufficient room the message "Overflow" appears. (NB At this stage it hasn't even started trying to fit the source in as well). During Pass 2 after each line is listed to the screen it is numbered and appended to the source file. At this point a check is done to ensure that the end of the file has not

started to overwrite the symbol table. If it has the "Overflow" message appears and dis-assembly stops. Although a partial source file exists at this time, one ZEAP parameter, the label count, will be incorrectly set.

The dis-assembler requires a little more memory to create the file than ZEAP does to hold it as the dis-assembler symbol table uses 3 bytes/entry while ZEAP uses only 2 bytes/entry.

If you wish to enter more than 36 address pairs in response to "DATA areas?" you will have to temporarily save ZEAP's workspace (fF04-fF2F) elsewhere and restore it afterwards. (See section 2).

6. USING A PRINTER

The "T" option causes the screen listing to be output to the serial port as well as to the screen, with a line feed output after every carriage return. As a result a serial printer, such as a Teletype, can be connected directly to the serial port. The characters are output through the SRLX routine within Nas-sys. (SCAL SRLX).

If you have a printer with a parallel interface connected to your system, then this can be driven through the Nas-sys User output routine. Either the control program can be altered so that when the "T" option is set the dis-assembler calls UOUT rather than SRLX, or the "U" command can be given in the monitor before running the dis-assembler. In the latter case the printed output will include all the program prompts and your replies.

APPENDIX A

Idiosyncrasies and Bugs

1. In common with a lot of software, if the dis-assembler is asked to dis-assemble code up to FFFF it will not stop at FFFF, but wraps round to 0 and will continue ad infinitum. The address to be entered to ensure that this does not happen is FFFF-length of instruction ending at FFFF. (eg FFFC if FFFD is the start of a JP instruction).
2. If you dis-assemble a very large program with option "P" set you will discover that the control program cannot count properly past page number 99. Page 100 appears as :0, page 110 as ;0, and so on. (See the ascii equivalents of 3A, 3B, 3C,).
3. If you dis-assemble an area of memory without setting option "U", (ie RCAL/SCAL/PRS assumed), then the following code sequences will not be detected as invalid code:-
..DD D7.../..DD DF.../..DD EF.../..FD D7.../..FD DF.../..FD EF..
The DD/FD prefix is ignored and the following code is treated as though it were a valid RCAL/SCAL/PRS instruction.

APPENDIX B

Modifications

Here is a short list of points in the program that you may wish to change to suit your own particular circumstances.

The number of printed lines per page is set at CB5A and is at present set to 3C (dec. 60).

The number of lines between pages is set at CB5B. This is actually set to one more than the number of lines wanted, and is presently set to 5 to give a gap of 4 lines.

(The total number of lines per page is 60+5+heading=66 (decimal)).

The end-of-line delay is set by the value at CFD7 which is at present set to 80 (128 decimal)).

The SRLX routine number (6F) is held at CFE3. This should be changed to 75 for UOUT (see section 6).

APPENDIX C

Writing your own control program for REVAS

The reverse assembler subroutine REVAS does the actual dis-assembly of the code in memory. The addition of labels, tape output, cross reference tables etc are functions performed by the external control routine. REVAS dis-assembles a specified block of memory, line by line, and after generating each line of source it calls an external output routine. In the case of the simple control program this just lists the line directly to the VDU. In the interactive control program two completely different "output" routines are used, depending on whether Pass 1 or Pass 2 is being executed.

REVAS requires three parameters to be passed to it on calling. These are set in the three register pairs BC DE HL and are:-

BC=The start address to be used on the listing.
DE=The start memory address of the code to be dis-assembled.
HL=The end memory address of the code to be dis-assembled.
(REVAS returns when this address is passed).

A jump to REVAS is held in the last three bytes of the program, and so you should CALL CFFD.

Before you call REVAS you have to set two items in its workspace. Firstly the variable RSTFLG (E03) has to be set. Bit 0 of it controls how REVAS handles RST instructions. If bit 0 is set then RCAL, SCAL and PRS are assumed, if bit 0 is clear then all RSTs are treated as single byte instructions with no following parameters. Secondly, when REVAS comes to output a line it sets register pair HL to point to the start of the output buffer and then calls address E00. Your program must set a JP instruction at this address to point to your "output" routine.

When REVAS calls the output routine there are several variables that could contain values of interest.

LINEA (EOE) holds the address of the instruction currently being listed.

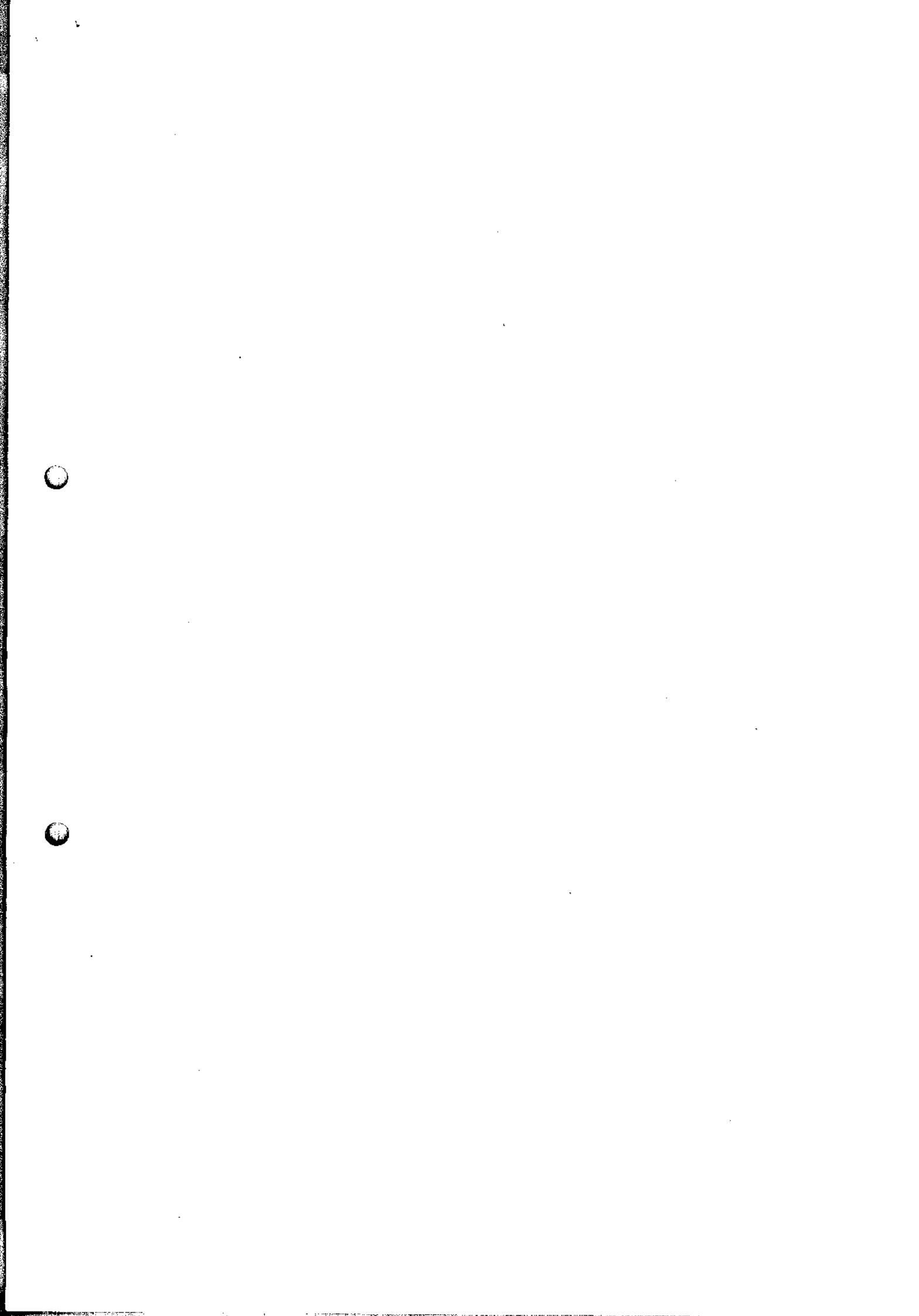
If the line contains a sixteen-bit operand, (ie the instruction is JP, JR, CALL, LD HL,... etc), LABELA (E10) holds its value and the variable LABELP (E12) points to the leading £ of the operand in the output buffer. If there is no sixteen-bit operand then LABELP is set to zero.

LABELP/LABELA/LINEA would be used by a program that generated labels.

For example:

Pass1:If LABELP is non-zero then LABELA holds a 16-bit reference to be entered to the symbol table, and LINEA holds the address for the cross-reference entry.

Pass2:Before outputting the line - check LINEA value and see if the line needs labelling. Also if LABELP is non-zero then a label needs to be set in the buffer at the point where LABELP is pointing.



APPENDIX D

Sample dis-assembly - Simple mode.

NB All user input is underlined.

address	code bytes	mnemonic	operand	character equivalents
<u>EC400 CDD6 2D</u>				
CDD6	CD E9 CD	CALL	#CDE9	;MiM
CDD9	36 00	LD	(HL),#00	;6.
CDDB	ED 5B 10 0E	LD	DE,(#0E10)	;m..
CDDF	2B	DEC	HL	;+
CDE0	73	LD	(HL),E	;s
CDE1	2B	DEC	HL	;+
CDE2	72	LD	(HL),D	;r
CDE3	2B	DEC	HL	;+
CDE4	CB 5F	BIT	3,A	;K_
CDE6	C8	RET	Z	;H
CDE7	18 CC	JR	#CDB5	;.L
CDE9	ED 5B 52 0E	LD	DE,(#0E52)	;m;R.
CDED	23	INC	HL	;z
CDEF	A7	AND	A	;
CDEF	06 00	LD	B,#00	;..
CDF1	ED 52	SBC	HL,DE	;mR
CDF3	EB	EX	DE,HL	;k
CDF4	ED 42	SBC	HL,BC	;mB
CDF6	42	LD	B,D	;B
CDF7	4B	LD	C,E	;K
CDF8	EB	EX	DE,HL	;k
CDF9	2A 4E 0E	LD	HL,(#0E4E)	;*N.
CDFC	2B	DEC	HL	;+
CDFD	ED 52	SBC	HL,DE	;mR
CDFF	30 0B	JR	NC,#CE0C	;0.
CE01	2A 52 0E	LD	HL,(#0E52)	;*R.
CE04	ED 53 52 0E	LD	(#0E52),DE	;mSR.
CE08	ED B0	LDIR		;m0
CE0A	2B	DEC	HL	;+
CE0B	C9	RET		;I
CE0C	EF	RST	#28	;o
CE0D	4F 76 65 72	DEFM	/Over/	;Over
CE11	66 6C 6F 77	DEFM	/flow/	;flow
CE15	20 61 74	DEFM	/ at/	; at
CE18	0D 00	DEFB	#0D,#00	;..
CE1A	21 14 0E	LD	HL,#0E14	;i..
CE1D	CD F3 CE	CALL	#CEF3	;MsN
CE20	DF	RST	#18	;
CE21	5B	DEFB	#5B	;i
CE22	01 13 00	LD	BC,#0013	;....
CE25	FD 21 58 0E	LD	IY,#0E58	;!x.
CE29	E5	PUSH	HL	;e
CE2A	DD E1	POP	IX	;i;a
CE2C	DD 09	ADD	IX,BC	;i.
CE2E	3A 44 0E	LD	A,(#0E44)	;:D.

APPENDIX E

Sample output - Interactive mode

N8 All user input is underlined.

EC400

Options? (STZXLPDHU)-LXP

Symbol table area?

1000 1FFF

Title?

Demonstration

What on?

CDD6 CE2E

DATA areas?

Go?

address	code bytes	label	mne.	operand	character equivalents
PAGE 01 Demonstration					
		L0013	EQU	\$0013	
		LOE10	EQU	\$0E10	
		LOE14	EQU	\$0E14	
		LOE44	EQU	\$0E44	
		LOE4E	EQU	\$0E4E	
		LOE52	EQU	\$0E52	
		LOE58	EQU	\$0E58	
		LCDB5	EQU	\$2CDB5	
CDD6	CD E9 CD		CALL	LCDE9	;MIM
CDD9	36 00		LD	(HL),E00	;6.
CDDB	ED 5B 10 0E		LD	DE,(LOE10)	;m1..
CDDF	2B		DEC	HL	;+
CDE0	73		LD	(HL),E	;S
CDE1	2B		DEC	HL	;+
CDE2	72		LD	(HL),D	;R
CDE3	2B		DEC	HL	;+
CDE4	CB 5F		BIT	3,A	;K
CDE6	C8		RET	Z	;H
CDE7	18 CC		JR	LCDB5	;L
CDE9	ED 5B 52 0E	LCDE9	LD	DE,(LOE52)	;m;R.
CDED	23		INC	HL	;+
CDEF	A7		AND	A	;
CDEF	06 00		LD	B,E00	;..
CDF1	ED 52		SBC	HL,DE	;mR
CDF3	EB		EX	DE,HL	;K
CDF4	ED 42		SBC	HL,BC	;mB
CDF6	42		LD	B,D	;B
CDF7	4B		LD	C,E	;K
CDF8	EB		EX	DE,HL	;K
CDF9	2A 4E 0E		LD	HL,(LOE4E)	;mN.
CDFC	2B		DEC	HL	;+
CDFD	ED 52		SBC	HL,DE	;mR
CDFF	30 0B		JR	NC,LCB0C	;O.
CE01	2A 52 0E		LD	HL,(LOE52)	;mR.

Unresolved label refs
outside the range of
code dis-assembled.

CE04	ED 53 52 0E	LD (LOE52),DE	;mSR.
CE08	ED B0	LDIR	;m0
CEOA	2B	DEC HL	;+
CEOB	C9	RET	;I
CEO C	EF	LCEO RST \$28	;o
CEOD	4F 76 65 72	DEFM /Over/	;Over
CE11	66 6C 6F 77	DEFM /flow/	;flow
CE15	20 61 74	DEFM / at/	; at
CE18	0D 00	DEFB \$0D,\$00	;..
CE1A	21 14 0E	LD HL,LOE14	;... ;MsN
CE1D	CD F3 CE	CALL LCEF3	
CE20	DF	RST \$18	;_
CE21	5B	DEFB \$5B	;i
CE22	01 13 00	LD BC,LOO13	;... ;.IX.
CE25	FD 21 58 0E	LD IY,LOE58	
CE29	E5	PUSH HL	;e
CE2A	DD E1	POP IX	;ja
CE2C	DD 09	ADD IX,BC	;j.
CE2E	3A 44 0E	LD A,(LOE44)	;:D.
LCEF3 EQU SCEF3			

PAGE 02 Demonstration

0013 CE22
 0E10 CDBB
 0E14 CE1A
 0E44 CE2E
 0E4E CDF9
 0E52 CDE9 CEO1 CEO4
 0E58 CE25
 CDB5 CDE7
 CDE9 CDD6
 CEOC CDFF
 CEF3 CE1D
 NAS-SYS 1

↑ ↑
address Location of references to the address

Simple Control Program for REVAS

```

C489      9992 ;*****+
C489      9993 ; SIMPLE CONTROL PROGRAM FOR REVAS 21-92-89
C489      9994 ; ENTERED BY "E" COMMAND
C489      9995 ;
C489      9996 ; THREE POSSIBLE FORMATS -
C489      9997 ; < E NNNN > - TRANSFERS CONTROL TO REVASC
C489      9998 ; < E NNNN AAAA > - STARTS SIMPLE DISASSEMBLY AT LINE AAAA
C489      9999 ; < E NNNN AAAA BB > - AS ABOVE BUT DISPLAYS
C489      9910 ; < BB > LINES AT A TIME
C489      9911 ;*****+
C489 3E C3 9912 LD A,$C3 ;SET RAM JUMP
C489 32 09 9E 9913 LD (PATCH),A
C489 32 03 9E 9914 LD (RSTFLG),A ;SET RSTFLG=NAS-SYS
C489 21 28 C4 9915 LD HL,PRINT ;PATCH ROUTINE
C489 22 91 9E 9916 LD (PATCH+1),HL
C489 3A 99 9C 9917 LD A,(NUMARG) ;CHECK OPTIONS
C411 FE 02 9918 CP 2 ;TWO?
C413 DA BB CA 9919 JP C,REVASC ;ONE,GO TO REVASC
C416 DF 9920 RST SCAL ;GET ARGUMENTS TO REGS.
C417 60 9921 DEFB ARGS
C418 29 #2 9922 JR NZ,#+2 ;SKIP IF 3 ENTERED
C41A 9E 91 9923 LD C,1 ;SET DEFAULT
C41C 41 9924 LD B,C ;SET COUNT
C41D ED 43 10 9C 9925 LD (ARG3),BC ;SAVE
C421 42 9926 LD B,D ;GET START ADDRESS...
C422 4B 9927 LD C,E ;...TO BC AS WELL
C423 21 FF FF 9928 LD HL,-1 ;SET END ADDRESS
C424 10 16 9929 JR REVAS ;GO TO REVAS
C428 9930 ;
C428 9931 ; PRINT (HL) UNTIL A CR, THEN CHECK LINE COUNT
C428 9932 ;
C428 7E 9933 PRINT: LD A,(HL) ;GET BYTE
C429 F7 9934 RST $30 ;OUTPUT A CHARACTER
C42A 23 9935 INC HL
C42B FE 99 9936 CP CR ;WAS IT A CR?
C42D 29 F9 9937 JR NZ,PRINT ;NO,CONTINUE
C42F 21 10 9C 9938 LD HL,ARG3 ;CHECK COUNT
C432 35 9939 DEC (HL) ;DONE?
C433 C8 9940 RET NZ ;NO
C434 23 9941 INC HL ;YES,RESET COUNT
C435 7E 9942 LD A,(HL)
C436 2B 9943 DEC HL
C437 77 9944 LD (HL),A
C438 CF 9945 RST RIN ;WAIT FOR KEY
C439 D6 19 9946 SUB ESC ;FINISH?
C43B C9 9947 RET NZ ;NO,CONTINUE
C43C DF 9948 RST SCAL ;YES,DONE
C43D 5B 9949 DEFB RETURN
C43E 9950 ;*****+
C43E 9951 ; EQUATES TO NAS-SYS ROUTINES
C43E 9952 ;*****+
9953 RIN: EQU 0
9954 SCAL: EQU $18
9955 RETURN: EQU $5B
9956 ARGS: EQU $60
$C08 9957 NUMARG: EQU $C08
$C19 9958 ARG3: EQU $C19
$913 9959 ESC: EQU $1B
9960 CR: EQU $D

```

REVAS subroutine version N1.1 88-93-89

```

C43E      9962 ;*****+
C43E      9963 ; A REVERSE ASSEMBLER FOR 8080/Z80 CODES
C43E      9964 ;
C43E      9965 ; WRITTEN BY DAVID PARKINSON
C43E      9966 ; VERSION N1.1 88-93-89
C43E      9967 ;
C43E      9968 ; THE OUTPUT IS IN ZEAP COMPATIBLE MNEMONICS
C43E      9969 ;
C43E      9970 ; THE PROGRAM IS IN THE FORM OF A SUBROUTINE
C43E      9971 ; REVAS IS ENTERED WITH THE FOLLOWING SEQUENCE....
C43E      9972 ;
C43E      9973 ; LD BC WITH THE "PROGRAM" ADDRESS
C43E      9974 ; (THE STARTING ADDRESS USED IN THE LISTING)
C43E      9975 ; LD DE WITH THE START MEMORY ADDRESS
C43E      9976 ; (ACTUAL LOCATION OF CODE TO BE REVERSE ASSEMBLED)
C43E      9977 ; LD HL WITH THE END ADDRESS
C43E      9978 ; (REVAS RETURNS WHEN THIS ADDRESS IS PASSED)
C43E      9979 ; CALL REVAS
C43E      9980 ;*****+
C43E      9981 ; VARIABLE STORAGE
C43E      9982 ;*****+
C43E      C43E 9983 SAVE: EQU *
C43E      9984 ORG $E9B ;WORKSPACE
C43E      9985 PATCH: DEFS 3 ;RAM PATCH TO ROUTINE
C43E      9986 RSTFLG: DEFS 1 ;RST HANDLING FLAG
C43E      9987 ADDR: DEFS 2 ;PROGRAM COUNTER
C43E      9988 HADDR: DEFS 2 ;CURRENT MEMORY ADDRESS
C43E      9989 EADDR: DEFS 2 ;WHERE TO STOP
C43E      9990 OPCADR: DEFS 2 ;POINTER IN BUFFR
C43E      9991 HXYFLG: DEFS 2 ;HL,IX,IY FLAG
C43E      9992 LINEA: DEFS 2 ;ADDR. LINE REF. LABEL
C43E      9993 LABELA: DEFS 2 ;LABEL VALUE
C43E      9994 LABELP: DEFS 2 ;POINTER TO POS. IN BUFFER
C43E      9995 BUFFER: DEFS 48 ;OUTPUT BUFFER
C43E      9996 ;*****+
C43E      9997 ; REVERSE ASSEMBLER ENTRY POINT
C43E      9998 ;*****+
C43E      9999 ORG SAVE
C43E      C43E 9100 REVAS: LD (ADDR),BC ;SAVE DATA
C442      EB 33 96 9E 9101 LD (HADDR),DE
C443      22 68 9E 9102 LD (EADDR),HL
C449      9103 ; NEXT LINE OF OUTPUT
C449      2A 98 9E 9104 NEXTL: LD HL,(EADDR) ;CHECK FOR END
C44C      ED 3B 96 9E 9105 LD DE,(HADDR) ;CLEAR CARRY
C459      AF 9106 XOR A
C451      ED 52 9107 SBC HL,DE ;RETURN IF PAST
C453      D8 9108 RET C
C454      CD 6E C4 9109 CALL INITB ;INITIALISE OUTPUT BUFFER
C457      CD 88 C4 9110 CALL BYTE ;GET NEXT BYTE
C458      CD 15 C5 9111 CALL DECODE ;DECODE IT
C45B      21 9C 9E 9112 LD HL,HXYFLG ;CHECK IF SET THEN USED
C468      7E 9113 LD A,(HL)
C461      23 9114 INC HL
C462      3E 9115 CP (HL)
C463      C4 53 C9 9116 CALL NZ,NOTVAL ;INVALID IF UNUSED
C466      21 14 9E 9117 LD HL,BUFFER ;SET ADDRESS
C469      CB 99 9E 9118 CALL PATCH ;PRINT OR WHATEVER
C46C      18 DB 9119 JR NEXTL ;LOOP
C46E      9120 ;*****+
C46E      9121 ; INITIALISE THE OUTPUT BUFFER (AND CERTAIN VARIABLES)

```

REVAS subroutine version N1.1 08-03-88

```

C46E      9122 ;*****+
C46E 11 14 0E 9123 INITB: LD DE,BUFFER ;GET START ADDRESS
C471 DS    9124 PUSH DE ;GET TO IX
C472 DD E1  9125 POP IX
C474 21 00 09 9126 LD HL,0 ;CLEAR LABEL POINTER
C477 22 12 0E 9127 LD (LABELP),HL ;CLEAR FLAG
C47A 22 0C 0E 9128 LD (HXYFLG),HL ;CLEAR FLAG
C47D 2A 04 0E 9129 LD HL,(ADDR) ;WRITE ADDRESS
C480 22 0E 0E 9130 LD (LINEA),HL ;KEEP ADDRESS
C483 CD 9D C4 9131 CALL HEX4
C486 E3    9132 EX DE,HL ;NOW CLEAR REST OF BUFFER
C487 23    9133 INC HL
C488 22 9A 0E 9134 LD (OPCADR),HL ;SET POINTER FOR OPCODES
C48B 23    9135 DEC HL
C48C 06 28  9136 LD B,43
C48E 36 29  9137 LD (HL),'
C490 23    9138 INC HL
C491 19 FB  9139 DJNZ #-5
C493 DD 36 2A 3B 9140 LD (IX+42),';COMMENT DELIM.
C497 36 0D    9141 LD (HL),CR ;SET CR AT END
C499 11 2D 0E 9142 LD DE,BUFFER+25 ;SET ADDRESS FOR MNEMONIC
C49C C9    9143 RET ;DONE
C49D      9144 ;*****+
C49D      9145 ;UTILITY SUBROUTINES
C49D      9146 ;*****+
C49D      9147 ;
C49D      9148 ;HL TO BUFFER AS FOUR HEX CHARACTERS
C49D      9149 ;
C49D 7C    9150 HEX4: LD A,H ;GET HI BYTE
C49E CD A2 C4 9151 CALL HEX2 ;PRINT IT
C4A1 7B    9152 LD A,L ;GET LO BYTE
C4A2      9153 ;TRANSFER A AS TWO HEX CHARACTERS TO
C4A2      9154 ;OUTPUT BUFFER (DE POINTS TO CURRENT POSITION
C4A2      9155 ;IN THE BUFFER).
C4A2      9156 ;
C4A2 F5    9157 HEX2: PUSH AF ;SAVE A
C4A3 0F    9158 RRCA ;SHIFT HI' NIBBLE DOWN.
C4A4 0F    9159 RRCA
C4A5 0F    9160 RRCA
C4A6 0F    9161 RRCA
C4A7 CD AB C4 9162 CALL HEX1 ;PRINT IT
C4AA F1    9163 POP AF ;RECOVER A
C4AB F3    9164 HEX1: PUSH AF ;SAVE A
C4AC E6 0F  9165 AND $0F ;ISOLATE LO' NIBBLE
C4AE C6 90  9166 ADD $90 ;MAKE ASCII
C4B0 27    9167 DAA
C4B1 CE 48  9168 ADC $48
C4B3 27    9169 DAA
C4B4 12    9170 LD (DE),A ;PUT IN BUFFER
C4B5 13    9171 INC DE ;BUMP ADDRESS
C4B6 F1    9172 POP AF ;RECOVER A
C4B7 C9    9173 RET
C4B8      9174 ;
C4B8      9175 ;GET NEXT BYTE OF DATA. PUT IN OUTPUT
C4B8      9176 ; STRING ALONG WITH ANY ASCII
C4B8      9177 ; EQUIVALENT
C4B8      9178 ;
C4B8 DD 23  9179 BYTE: INC IX ;BUMP POINTER
C4B9 2A 04 0E 9180 LD HL,(ADDR) ;UPDATE PC
C4B0 23    9181 INC HL

```

REVAS subroutine version N1.1 02-03-88

C4BE 22 94 9E	9182	LD	(ADDR),HL	
C4C1 2A 94 9E	9183	LD	HL,(MADDR)	;LOAD BYTE ADDRESS
C4C4 7E	9184	LD	A,(HL)	;LOAD BYTE
C4C5 23	9185	INC	HL	;UPDATE ADDRESS
C4C6 22 94 9E	9186	LD	(MADDR),HL	
C4C9 2A 9A 9E	9187	LD	HL,(OPCADR)	;NOW PUT IN OUTPUT...
C4CC 23	9188	INC	HL	;SPACE BETWEEN OPCODES
C4CD EB	9189	EX	DE,HL	;..STRING IN CORRECT..
C4CE CD A2 C4	9190	CALL	HEX2	;..PLACE.
C4D1 EB	9191	EX	DE,HL	;RESTORE POINTER
C4D2 22 9A 9E	9192	LD	(OPCADR),HL	;SAVE IN CASE NEEDED AGAIN
C4D5 F5	9193	PUSH	AF	;SAVE
C4D6 3C	9194	INC	A	;ROLL "RUB OUT" ROUND
C4D7 E6 7F	9195	AND	\$7F	;STRIP OFF PARITY
C4D9 FE 21	9196	CP	\$21	;PRINTABLE?
C4DB 3D	9197	DEC	A	;((CORRECT A))
C4DC 39 92	9198	JR	NC,*+2	;YES, SKIP
C4DE 3E 2E	9199	LD	A,'.'	;NO, SET ''
C4E0 DD 77 2A	9200	LD	(IX+42),A	;APPEND TO BUFFER
C4E3 F1	9201	POP	AF	;RECOVER A
C4E4 C9	9202	RET		
C4E5	9203 ;			
C4E5	9204 ; WRITE "EX"			
C4E5	9205 ;			
C4E5 EB	9206 WREX:	EX	DE,HL	
C4E6 36 45	9207	LD	(HL),'E'	
C4E8 23	9208	INC	HL	
C4E9 36 58	9209	LD	(HL),'X'	
C4EB 18 96	9210	JR	URLD9	;REST IN URLD
C4ED	9211 ;			
C4ED	9212 ;WRITE "LD"			
C4ED	9213 ;			
C4ED EB	9214 URLD:	EX	DE,HL	
C4EE 36 4C	9215	LD	(HL),'L'	
C4F0 23	9216	INC	HL	
C4F1 36 44	9217	LD	(HL),'D'	
C4F3 11 32 9E	9218 URLD9:	LD	DE,BUFFER+J8 ;SET OPERAND ADDRESS	
C4F6 C9	9219	RET		
C4F7	9220 ;			
C4F7	9221 ; WRITE ','			
C4F7	9222 ;			
C4F7 EB	9223 COMMA:	EX	DE,HL	
C4F8 36 2C	9224	LD	(HL)','	
C4FA 23	9225	INC	HL	
C4FB EB	9226	EX	DE,HL	
C4FC C9	9227	RET		
C4FD	9228 ;			
C4FD	9229 ;WRITE "**"			
C4FD	9230 ;			
C4FD EB	9231 POUND:	EX	DE,HL	
C4FE 36 23	9232	LD	(HL),'#'	
C500 23	9233	INC	HL	
C501 EB	9234	EX	DE,HL	
C502 C9	9235	RET		
C503	9236 ;			
C503	9237 ;COPY OVER CHARACTERS			
C503	9238 ;			
C503 ED A9	9239 COPY6:	LDI		
C505 ED A9	9240 COPY5:	LDI		
C507 ED A9	9241 COPY4:	LDI		

REVAS subroutine version N1.1 08-03-89

C589 ED A8	0242 COPY3: LDI
C59B ED A8	0243 COPY2: LDI
C58D ED A8	0244 LDI
C59F C9	0245 RET
C518	0246 ;
C519	0247 ;FORM ADDRESS FOR TABLE LOOK UP
C519	0248 ;
C519 85	0249 FTABR: ADD L
C511 6F	0250 LB L,A
C512 D8	0251 RET NC
C513 24	0252 INC H
C514 C9	0253 RET
C515	0254 ;*****
C515	0255 ; PRELIMINARY DECODE OF BYTE
C515	0256 ;*****
C515 F3	0257 DECODE: PUSH AF ;SAVE
C516 E6 C8	0258 AND \$C8 ;ISOLATE MSBS
C518 FE 48	0259 CP \$48 ;#1XXXXXX?
C51A CA 90 C5	0260 JP Z,LOAD8 ;YES, 8-BIT LOAD
C51B FE 80	0261 CP \$80 ;#0XXXXXX?
C51F CA 94 C6	0262 JP Z,ARITH8 ;YES, 8-BIT ARITHMETIC
C522 F1	0263 POP AF ;GET BYTE BACK
C523 F3	0264 PUSH AF ;PUT ON STACK FOR LATER
C524 E6 8F	0265 AND \$8F ;ISOLATE 7XXX3210
C526 #7	0266 RLCA ;FORM AS ..321#7.
C527 #7	0267 RLCA
C528 21 7F C9	0268 LD HL,TABLE ;LOAD BASE ADDRESS
C529 CD 19 C5	0269 CALL FTABR ;FORM ADDRESS
C52E 7E	0270 LD A,(HL) ;GET ADDRESS AT THAT ADDRESS
C52F 23	0271 INC HL
C530 66	0272 LD H,(HL)
C531 6F	0273 LD L,A
C532 E9	0274 JP (HL) ;GO TO IT
C533	0275 ;*****
C533	0276 ; DECODE OPCODES FULLY
C533	0277 ;*****
C533	0278 ;
C533	0279 ;INC/DEC REGISTER/REGISTER PAIR
C533	0280 ;*****
C533 21 BF C9	0281 INC: LD HL,INCM ;POINT TO "INC"
C534 18 #3	0282 JR **3 ;SKIP
C538 21 C2 C9	0283 DEC: LD HL,DECM ;POINT TO "DEC"
C538 CD 38 C7	0284 CALL COPY3S ;COPY OVER
C53E F1	0285 POP AF ;RECOVER OPCODE
C53F CB 57	0286 BIT 2,A ;TEST 8/16 BIT
C541 28 34	0287 JR Z,REGPR ;JUMP IF 16 BIT
C543 #F	0288 RRCA ;SHIFT REG. ID...
C544 #F	0289 RRCA ;...DOWN TO LOW A
C545 #F	0290 RRCA
C546 18 6C	0291 JR SREG ;RETURN AFTER PRINTING REGISTER
C548	0292 ;*****
C548	0293 ;16-BIT DIRECT LOAD
C548	0294 ;*****
C548 CB ED C4	0295 LD16: CALL WRLD ;WRITE "LD"
C548 F1	0296 POP AF ;RECOVER OPCODE
C54C CD 77 C5	0297 CALL REGPR ;PRINT REGISTER PAIR
C54F CB F7 C4	0298 CALL COMMA ;","
C552 CD BB C4	0299 LD16A: CALL BYTE ;GET LO BYTE
C555 4F	0300 LD C,A ;SAVE IN C
C556 CD BB C4	0301 CALL BYTE ;GET HI BYTE

REVAS subroutine version N1.1 98-93-88

C559 67	0302	LD H,A	;PUT IN H
C55A 69	0303	LD L,C	;RECOVER FIRST BYTE
C55B ED 33 12 9E	0304 LD16B:	LD (LABELP),DE	;SAVE POINTER
C55F 22 19 9E	0305	LD (LABELA),HL	;SAVE ADDRESS (?)
C562 CD FD C4	0306	CALL POUND	";"
C563 C3 9D C4	0307	JP HEX4	;RETURN,PRINTING HL.
C568	0308 ;*****		
C568	0309 ; ADD HL		
C568	0310 ;*****		
C568 21 08 C9	0311 ADDHL:	LD HL,ARTAB	;POINT TO "ADD"
C568 CD 38 C7	0312	CALL COPY3S	;WRITE IT
C56E 3E 29	0313	LD A,\$29	;MAKE LIKE "HL"
C578 CD 77 C5	0314	CALL REGPR	;WRITE HL,IX,OR IY
C573 CD F7 C4	0315	CALL COMMA	";"
C576 F1	0316	POP AF	;RECOVER OPCODE
C577	0317 ;*****		
C577	0318 ;REGISTER PAIR DECODE/PRINT		
C577	0319 ;*****		
C577 47	0320 REGPR:	LD B,A	;SAVE IN B
C578 9F	0321	RRCA	;SHIFT PAIR ID BITS DOWN.
C579 9F	0322	RRCA	
C57A 9F	0323	RRCA	
C57B C6 02	0324	ADD 2	;ALTER REG. CODING
C57D 0E 04	0325	LD C,6	;PUT 6 IN C
C57F A1	0326	AND C	;ISOLATE REG. ID
C580 21 C8 C9	0327	LD HL,RPTTAB	;LOAD BASE ADDRESS
C583 29 98	0328	JR NZ,NOTSP	;SKIP IF NOT 'SP'
C585 29	0329	DEC HL	;POINT TO "SP"
C586 C8 78	0330	BIT 7,B	;IS IT PUSH/POP?
C588 CA 9B C5	0331	JP Z,COPY2	;NO DONE
C589 29	0332	DEC HL	;YES,POINT...
C58C 29	0333	DEC HL	;...TO "AF"
C58D B9	0334 NOTSP:	CP C	;IS IT "HL"?
C58E 29 97	0335	JR NZ,RP2	;NO,SKIP
C594 3A 9C 9E	0336	LD A,(HXYFLG)	;LOAD FLAG
C593 32 9D 9E	0337	LD (HXYFLG+1),A	;SET 'USED'
C596 81	0338	ADD C	;ADJUST IF IX/IY
C597 CD 18 C5	0339 RP2:	CALL FTADR	;FORM ADDRESS
C59A C3 9B C5	0340	JP COPY2	;COPY OVER
C59D	0341 ;*****		
C59D	0342 ;8-BIT LOAD INSTRUCTIONS		
C59D	0343 ;*****		
C59D F1	0344 LOAD8:	POP AF	;RECOVER OPCODE
C59E FE 76	0345	CP #76	;HALT?
C5A0 28 5C	0346	JR Z,HALT	;YES,JUMP
C5A2 F5	0347	PUSH AF	;SAVE OPCODE
C5A3 CD ED C4	0348	CALL WRLD	;WRITE "LD"
C5A6 9F	0349	RRCA	;SHIFT REG ID DOWN
C5A7 9F	0350	RRCA	
C5A8 9F	0351	RRCA	
C5A9 CD B4 C5	0352	CALL SREG	;PRINT IT
C5AC CD F7 C4	0353	CALL COMMA	";"
C5AF F1	0354 L8B:	POP AF	;GET OPCODE BACK
C5B0 CB 77	0355	BIT 6,A	;TEST FOR IMMED. DATA
C5B2 29 41	0356	JR Z,INN	;JUMP IF SO
C5B4	0357 ;*****		
C5B4	0358 ;SINGLE REGISTER DECODE		
C5B4	0359 ;*****		
C5B4 3C	0360 SREG:	INC A	;ADJUST CODING
C5B5 E6 97	0361	AND 7	;ISOLATE IT

REVAS subroutine version M1.1 #8-#3-88

C5B7 FE 87	9362	CP	7	;IS IT MEMORY?
C5B9 28 99	9363	JR	Z,HEM	;YES, JUMP
C5B9 21 C9 C9	9364	LB	HL,RPTTAB+1	;LOAD BASE ADDRESS
C5B9 CD 19 C3	9365	CALL	FTADR	;FORM ADDRESS
C5C1 ED A8	9366	LBI		;COPY OVER REG
C5C3 C9	9367	RET		
C5C4 3E 28	9368	MEM:	LD A,(;HEM INDIRECT
C5C4 12	9369		LD (BE),A	;WRITE "("
C5C7 13	9370	INC	DE	;BUMP ADDRESS
C5C8 21 CE C9	9371	LB	HL,HXTTAB	
C5C8 3A 9C 9E	9372	LD	A,(HXYFLG)	;OFFSET IF INDEX
C5C8 32 9B 9E	9373	LD	(HXYFLG+1),A	;SET 'USED FLAG'
C5D1 F3	9374	PUSH	AF	;SAVE
C5D2 CD 97 C3	9375	CALL	RP2	;COPY REG. PR OVER
C5D5 F1	9376	POP	AF	;GET IT BACK
C5D6 A7	9377	AND	A	;IS IT INDEX?
C5D7 28 17	9378	JR	Z,NOTIXY	;NO, JUMP
C5D9	9379	:		
C5D9 CD BB C4	9380	CALL	BYTE	;YES, GET OFFSET
C5DC A7	9381	AND	A	;TEST IT
C5D9 9E 28	9382	LD	C,+'	;SET POSITIVE
C5DF 28 9F	9383	JR	Z,NOTIXY	;SKIP IF ZERO
C5E1 F2 E8 C5	9384	JP	P,PLUS	;JUMP IF IT WAS +VE
C5E4 ED 44	9385	NEG		;2'S COMP
C5E6 9E 29	9386	LD	C,'-'	;CHANGE SIGN
C5E8 47	9387	PLUS:	LD B,A	;SAVE MAGNITUDE
C5E9 79	9388	LD	A,C	;GET SIGN
C5EA 12	9389	LD	(DE),A	;PRINT IT
C5EB 13	9390	INC	DE	;BUMP ADDRESS
C5EC 78	9391	LD	A,B	;GET MAGNITUDE
C5ED CD FB C5	9392	CALL	PHEX2	;WRITE "XXX"
C5F0 3E 29	9393	NOTIXY:	LD A,")"	;CLOSE BRACKETS
C5F2 12	9394	LD	(DE),A	
C5F3 13	9395	INC	DE	
C5F4 C9	9396	RET		
C5F3	9397	*****		
C5F5	9398	*****	GET SINGLE BYTE IMMEDIATE	
C5F5	9399	*****		
C5F5 CD BB C4	9400	IMM:	CALL ,BYTE	;GET DATA
C5F8 CD F9 C4	9401	PHEX2:	CALL POUND	;PRINT ""
C5FB C3 A2 C4	9402		JP HEX2	;PRINT DATA
C5FE	9403	*****		
C5FE	9404	*****	HALT INSTRUCTION	
C5FE	9405	*****		
C5FE 21 D4 C9	9406	HALT:	LB HL,HALTH	
C601 C3 97 C5	9407		JP COPY4	;EXIT VIA COPY4
C604	9408	*****		
C604	9409	*****	8-BIT ARITHMETIC	
C604	9410	*****		
C604 F1	9411	ARITHB:	POP AF	;RECOVER OPCODE
C605 EE 48	9412		XOR \$40	;TOGGLE BIT
C607 F5	9413		PUSH AF	;PUT IT BACK
C608 9F	9414		RRCA	;GET OP ID
C609 9F	9415		RRCA	
C60A E6 9E	9416		AKD \$E	;ISOLATE IT
C60C 47	9417		LB B,A	;S3 IT
C60D 9F	9418		RRCA	
C60E 88	9419		ADD B	
C60F 21 D8 C9	9420		LB HL,ARTAB	;LOAD BASE ADDRESS
C612 C4 19 C5	9421		CALL NZ,FTADR	;FORM ADDRESS

REVAS subroutine version N1.1 #8-93-88

C615 78	9422	LD A,B	;RELOAD OP ID
C616 CD 38 C7	9423	CALL COPY3S	;COPY MNEMONIC
C619 21 98 CA	9424	LD HL,AQCOMM	;POINT TO "A,"
C61C FE 96	9425	CP 4	;SBC?
C61E CC 98 C5	9426	CALL Z,COPY2	;YES,DO IT
C621 FE 93	9427	CP 3	;ADD,ADCT
C623 DC 98 C5	9428	CALL C,COPY2	;YES,DO IT.
C626 18 87	9429	JR LBB	;PRINT OPERAND
9430 ;*****			
C628	9431 ; POP AND PUSH		
C628	9432 ;*****		
C628 21 F4 C9	9433 POP:	LD HL,POPM	;LOAD POINTER
C628 18 93	9434	JR **+3	;SKIP
C62D 21 F8 C9	9435 PUSH:	LD HL,PUSHM	;LOAD POINTER
C630 CD 97 C5	9436	CALL COPY4	;PRINT MNEMONIC
C633 13	9437	INC DE	;SPACE
C634 C3 76 C5	9438	JP REGPR-1	;GET OP & PR. REG PAIR
9439 ;*****			
C637	9440 ; CALL/IX/EXTENDED/IY		
C637	9441 ;*****		
C637 F1	9442 CALETc:	POP AF	;RECOVER OPCODE
C638 FE ED	9443	CP \$ED	;EXTENDED?
C63A CA C9 C8	9444	JP Z,EXTND	;YES,JUMP
C63D FE CD	9445	CP \$CD	;CALL?
C63F 28 17	9446	JR Z,CJR+1	;YES,JUMP
C641	9447 ; SET IX/IY FLAG		
C641 D6 F9	9448	SUB \$F9	;LEAVES 4 IF IY
C643 39 92	9449	JR NC,*+2	;SKIP IF 4
C643 JE 92	9450	LD A,2	;ELSE LOAD 2
C647 32 9C 9E	9451	LD (HXYFLG),A	;SET FLAG
C64A CD 98 C4	9452	CALL BYTE	;GET NEXT BYTE
C64D 47	9453	LD B,A	;TEMP SAVE
C64E E6 9F	9454	AND \$F	;CHECK LO' NIBBLE
C650 FE 9D	9455	CP \$D	;IS IT -D?
C652 78	9456	LD A,B	;RESET OPCODE)
C653 C8	9457	RET Z	;YES,INVALID CODE
C654 C3 15 C5	9458	JP DECODE	;PERHAPS NOT,DECODE IT
C657	9459 ;		
C657	9460 ; CALL/JUMP/RETURN		
C657	9461 ;		
C657 F1	9462 CJR:	POP AF	;RECOVER OPCODE
C658 F5	9463	PUSH AF	;SAVE IT
C659 E6 96	9464	AND 6	;ISOLATE ID
C659 87	9465	RLCA	
C65C 21 F8 C9	9466	LD HL,CJRTAB	;LOAD BASE ADDRESS
C65F C4 10 C5	9467	CALL NZ,FTADR	;FORM ADDRESS
C662 CD 97 C5	9468	CALL COPY4	;WRITE MNEMONIC
C665 13	9469	INC DE	;SPACE
C666 F1	9470	POP AF	;GET OPCODE BACK
C667 FE C9	9471	CP \$C9	;UNCOND RETURN?
C669 C8	9472	RET Z	;YES,RETURN
C66A C9 47	9473	BIT 9,A	;UNCOND. CALL/JUMP?
C66C 28 93	9474	JR NZ,UNCND	;YES,SKIP
C66E F3	9475	PUSH AF	;SAVE AGAIN
C66F CD 7C C6	9476	CALL CCODES	;ADD CONDITION
C672 F1	9477	POP AF	;RECOVER AGAIN
C673 E6 97	9478	AND 7	;CHECK FOR RETURN?
C675 C8	9479	RET Z	;YES,RETURN
C676 CD F7 C4	9480	CALL COMMA	";"
C679 C3 52 C5	9481 UNCOND:	JP LD16A	;EXIT WRITING "NNNN"

REVAS subroutine version N1.1 98-03-09

C67C 9482 ;
C67C 9483 ; DECODE CONDITION CODES
C67C 9484 ;
C67C 0F 9485 CCODES: RRCA ;SHIFT DOWN
C67D 0F 9486 RRCA
C67E E6 9E 9487 AND SE ;ISOLATE
C680 21 94 CA 9488 LD HL,CCTAB ;LOAD BASE ADDRESS
C683 C3 97 C5 9489 JP RP2 ;COPY OVER
C686 9490 ;*****
C686 9491 ; SINGLE BYTE ROTATE & MISC.
C686 9492 ;*****
C686 F1 9493 ROTNIS: POP AF ;RECOVER OPCODE
C687 0F 9494 RRCA
C688 E6 1C 9495 AND \$1C ;ISOLATE ID
C68A 21 14 CA 9496 LD HL,RMTAB ;LOAD BASE ADDRESS
C6BD C4 18 C5 9497 CALL NZ,FTADR ;FORM ADDRESS
C690 C3 97 C5 9498 JP COPY4 ;WRITE TO BUFFER
C699 9499 ;*****
C693 9500 ; RESTART
C693 9501 ;*****
C693 21 34 CA 9502 RST: LD HL,RSTM ;WRITE "RST"
C696 CD 38 C7 9503 CALL COPY3S
C699 F1 9504 POP AF ;RECOVER OPCODE
C69A E6 38 9505 AND \$38 ;ISOLATE ADDRESS
C69C CD F8 C3 9506 CALL PHEX2 ;WRITE "XXX"
C69F 21 83 9E 9507 LD HL,RSTFLG ;CHECK HANDLING FLAG
C6A2 CB 44 9508 BIT 0,(HL) ;NAS-SYS?
C6A4 CB 9509 RET Z ;RETURN IF NOT
C6A5 FE 10 9510 CP \$10 ;RCAL?
C6A7 D8 9511 RET C ;YES,DONE
C6A8 FE 29 9512 CP \$29 ;BRKPT?
C6AA C9 9513 RET Z ;YES,RETURN
C6AB 3B 3C 9514 JR C,FLUSH ;SKIP IF RCAL/SCAL
C6AD FE 28 9515 CP \$28 ;PRS?
C6AF CB 9516 RET NZ ;NO,RETURN
C6B0 9517 ; DECODE RST AS NASCON "PRS"
C6B0 9518 ; DECODE FOLLOWING STRING AS DEFN OR DEFB
C6B0 CD E9 C6 9519 BORN: CALL FLUSH ;PRINT "RST",GET NEXT BYTE
C6B3 06 93 9520 LD B,3 ;SET FOR 3 'DEFB'S
C6B5 38 22 9521 JR C,UNPRN ;SKIP IF DEFB
C6B7 94 9522 INC B ;4 IF 'DEFN'
C6B8 21 38 9E 9523 LD HL,BUFFER+2B ;SET HL
C6B8 36 40 9524 LD (HL),'N' ;OVERWRITE 'B' WITH 'N'
C6B0 23 9525 INC HL ;SPACE ON
C6BE 23 9526 INC HL
C6BF 36 2F 9527 LD (HL),// ;SET DELIMITER
C6C1 23 9528 INC HL
C6C2 EB 9529 EX DE,HL ;pointer to DE
C6C3 18 98 9530 JR HS1 ;SKIP ON
C6C5 CD FC C6 9531 MOREMS: CALL PRTABL ;NEXT BYTE PRINTABLE?
C6C8 38 97 9532 JR C,DONEN ;NO,PRINT LINE
C6CA CD B8 C4 9533 CALL BYTE ;YES,GET IT
C6CD 12 . 9534 HS1: LD (DE),A ;SET CHARAC. IN O/P
C6CE 13 9535 INC DE
C6CF 19 F4 9536 DJNZ MOREMS ;LOOP FOR MORE
C6D1 EB 9537 DONEN: EX DE,HL ;CLOSE STRING
C6D2 36 2F 9538 LD (HL),//
C6D4 18 DA 9539 JR BORN ;LOOP
C6D6 9540 ; DEFB SECTION
C6D6 CD B8 C4 9541 IN9: CALL BYTE ;GET NEXT BYTE

REVAS subroutine version N1.1 08-03-89

C4D9 B7	9542 UNPRN:	OR A	;CHECK FOR EOS
C6DA CA 53 C9	9543 JP Z,NOTVAL	;YES, EXIT THRU' NOTVAL	
C6DD CD FC C6	9544 CALL PRTABL	;NEXT BYTE PRINTABLE?	
C6E8 38 82	9545 JR NC,UN1	;NO, PRINT WHAT'S THERE	
C6E2 10 F2	9546 DJNZ UN#	;LOOP IF ROOM FOR MORE	
C6E4 CD 53 C9	9547 UN1: CALL NOTVAL	;PRINT LINE	
C6E7 18 C7	9548 JR BORN	;LOOP	
C6E9 21 14 9E	9549 ;PRINT BUFFER, GET NEXT BYTE & CHECK IF PRINTABLE		
C6EC CD 98 9E	9550 FLUSH: LD HL,BUFFER		
C6EF CD 46 C4	9551 CALL PATCH		
C6F2 CD 88 C4	9552 CALL INITB		
C6F3 F3	9553 CALL BYTE		
C6F6 CD 53 C9	9554 PUSH AF	;SAVE BYTE	
C6F9 F1	9555 CALL NOTVAL	;SET 'DEFB'	
C6FA 18 94	9556 POP AF	;RESET BYTE	
C6FC 24 96 9E	9557 JR PRTB1	;SKIP	
C6FF 7E	9558 PRTABL: LD HL,(ADDR)	;GET NEXT BYTE	
C700 FE 28	9559 LD A,(HL)		
C700 FE 28	9560 PRTB1: CP //	;=>SPACE?	
C702 08	9561 RET C	;NO, RETURN	
C703 FE 2F	9562 CP //	;BAR DEFN DELIMITER	
C705 37	9563 SCF		
C706 C8	9564 RET Z		
C707 FE 7F	9565 CP \$7F		
C709 3F	9566 CCF		
C70A C7	9567 RET		
C70B	9568 ;*****		
C70B	9569 ; NOP/DJNZ/JR NZ/JR NC		
C70B	9570 ;*****		
C70B F1	9571 NOPETC: POP AF	;RECOVER OPCODE	
C70C FE 18	9572 CP \$18	;DJNZ?	
C70E 28 2E	9573 JR Z,DJNZ	;YES, JUMP	
C710 38 23	9574 JR C,NOP	;JUMP ALSO IF<0.	
C712	9575 ;		
C712	9576 ; JUMP RELATIVE		
C712	9577 ;		
C712 EB	9578 JR: EX DE,HL	;WRITE "JR"	
C713 36 4A	9579 LD (HL),'J'		
C715 23	9580 INC HL		
C716 36 32	9581 LD (HL),'R'		
C718 11 32 9E	9582 LD DE,BUFFER+30	;RESET DE	
C71B FE 18	9583 CP \$18	;UNCONDITIONAL?	
C71D 28 88	9584 JR Z,UCD	;YES, JUMP	
C71F E6 18	9585 AND \$18	;ISOLATE COND. CODES	
C721 CD 7C C6	9586 CALL CCODES	;WRITE APPROPRIATE ONE	
C724 CD F7 C4	9587 CALL COMM	";,"	
C727 CD 88 C4	9588 UCD: CALL BYTE	;GET OFFSET	
C72A 4F	9589 LD C,A	;PUT IN C	
C72B #7	9590 RLCA	;EXTEND SIGN THRU' A	
C72C 9F	9591 SBC A		
C72D 47	9592 LD B,A	;PUT IN B	
C72E 2A 94 9E	9593 LD HL,(ADDR)	;LOAD CURRENT ADDRESS	
C731 #9	9594 ADD HL,BC	;COMPUTE TARGET ADDRESS	
C732 C3 5B C3	9595 JP LD16B	;WRITE IT ETC.	
C735	9596 ;		
C735	9597 ; WRITE "NOP"		
C735	9598 ;		
C733 21 37 CA	9599 NOP: LD HL,NOPM		
C738 CD 99 C5	9600 COPY3S: CALL COPY3	;COPY OVER MNEMONIC	
C738 13	9601 INC DE	;SPACE ON	

REVAS subroutine version N1.1 88-03-09

C73C 13	9682	INC DE
C73D C9	9683	RET
C73E	9684	;
C73E	9685	;DJNZ
C73E	9686	;
C73E 21 3A CA	9687	DJNZ: LD HL,DJNZM ;WRITE "DJNZ"
C741 CD 07 C3	9688	CALL COPY4
C744 13	9689	INC DE ;SPACE
C745 18 E8	9610	JR UCD ;REST IN "JR" ROUTINE
C747	9611	*****
C747	9612	;EX AF/AF'/JR/JR Z/JR C
C747	9613	*****
C747 F1	9614	EXAETC: POP AF ;RECOVER OPCODE
C748 FE 98	9615	CP 8 ;JR?
C74A 28 C6	9616	JR NZ,JR ;YES,GO TO THEM
C74C	9617	;
C74C CB E5 C4	9618	CALL WREX ;WRITE "EX"
C74F 21 3E CA	9619	LD HL,EXAFN ;WRITE "AF,AF"
C752 C3 93 C5	9620	JP COPY6
C753	9621	*****
C753	9622	;JP/OUT/EX (SP),HL/DI
C753	9623	*****
C755 F1	9624	JPETC: POP AF ;RECOVER OPCODE
C756 FE D3	9625	CP \$D3 ;JUMP?
C758 DA 58 C6	9626	JP C,CJR+1 ;JUMP IF "JP"
C75B CA EC C8	9627	JP Z,OUT ;JUMP IF "OUT"
C75E FE F3	9628	CP \$F3 ;"DI"?
C760 28 9E	9629	JR Z,DI ;YES,JUMP
C762	9630	;
C762 CD E5 C4	9631	CALL WREX ;WRITE "EX"
C763 21 44 CA	9632	LD HL,BSPBM ;WRITE "(SP),"
C768 CD 95 C5	9633	CALL COPY5
C768 3E 28	9634	LD A,\$28 ;SET FOR HL/IX/IY
C768 C3 77 C5	9635	JP REGPR ;WRITE APPROPRIATE ONE.
C770	9636	*****
C770	9637	; DI/EI
C770	9638	*****
C770 3E 44	9639	DI: LD A,'D' ;WRITE "DI"...
C772 18 92	9640	JR **2
C774 3E 45	9641	EI: LD A,'E' ;..OR "EI"
C776 EB	9642	EX DE,HL
C777 77	9643	LD (HL),A
C778 23	9644	INC HL
C779 36 49	9645	LD (HL),'I'
C77B C9	9646	RET
C77C	9647	*****
C77C	9648	;ROTATE & TEST/IN/EX DE,HL/EI
C77C	9649	*****
C77C F1	9650	CBETC: POP AF ;RECOVER OPCODE
C77D FE DB	9651	CP \$DB ;IN?
C77F CA A6 C8	9652	JP Z,IN ;YES,JUMP
C782 DA BD C8	9653	JP C,CD ;JUMP IF "CD"
C785 FE FB	9654	CP \$FB ;"EI"?
C787 28 EB	9655	JR Z,EI ;YES,JUMP
C789	9656	;EX DE,HL
C789 CD E5 C4	9657	CALL WREX ;WRITE "EX"
C78C 21 CC C9	9658	LD HL,RPTTAB+4 ;WRITE "DE"
C78F CD 9B C5	9659	CALL COPY2
C792 CD F7 C4	9660	CALL CONMA ;","
C793 C3 9B C5	9661	JP COPY2 ;WRITE "HL" & RETURN

REVAS subroutine version N1.1 08-93-89

```
C798          8662 ;*****  
C798          8663 ; RET/EXX/JP (HL)/LD SP,HL  
C798          8664 ;*****  
C798 F1          8665 RETTC: POP AF      ;RECOVER OPCODE  
C799 FE D9          8666 CP $D9      ;"RET" OR "EXX"?  
C798 DA 58 C6          8667 JP C,CJRTAB+1 ;"RET",SO JUMP  
C79E 28 1D          8668 JR Z,EXX     ;"EXX" SO JUMP  
C7A0 FE F9          8669 CP $FF      ;"LD SP"?  
C7A2 28 0B          8670 JR Z,LDSP    ;YES,JUMP  
C7A4 21 FC C9          8671 LD HL,CJRTAB+4 ;WRITE "JP"  
C7A7 CD 93 C5          8672 CALL COPY2   ;  
C7AA 21 32 0E          8673 LD HL,BUFFER+30 ;RESET HL  
C7AB 18 3D          8674 JR LD1+3    ;WRITE HL,IX OR IY  
C7AF          8675 ;  
C7AF          8676 ; LOAD SP  
C7AF          8677 ;  
C7AF CD ED C4          8678 LDSP: CALL WRLD    ;WRITE "LD"  
C7B2 21 4F CA          8679 LD HL,SPH    ;  
C7B5 CD 09 C5          8680 CALL COPY3   ;WRITE "SP"  
C7B8 3E 26          8681 LD A,$26    ;MAKE LIKE HL  
C7B9 C3 77 C5          8682 JP REGPR    ;WRITE HL,IX,OR IY  
C7BD          8683 ;  
C7BD          8684 ; WRITE "EXX"  
C7BD          8685 ;  
C7BD CD E5 C4          8686 EXX: CALL UREX    ;WRITE "EX"  
C7C8 23          8687 INC HL      ;BUMP ADDRESS  
C7C1 36 58          8688 LD (HL),'X'  ;ADD ANOTHER "X"  
C7C3 C9          8689 RET       ;  
C7C4          8690 ;*****  
C7C4          8691 ; LD (BC),A LD (DE),A LD(NN),HL LD,(NN),A  
C7C4          8692 ;*****  
C7C4 CD ED C4          8693 STIND: CALL WRLD    ;WRITE "LD"  
C7C7 F1          8694 POP AF      ;RECOVER OPCODE  
C7C8 F5          8695 PUSH AF     ;SAVE IT FOR ST16I  
C7C9 FE 22          8696 CP $22      ;16 BIT?  
C7CB 28 20          8697 JR Z,ST16I  ;YES,JUMP  
C7CD F1          8698 POP AF      ;NO,THROW AWAY SAVE  
C7CE CD E9 C7          8699 CALL LD1     ;WRITE INDIRECT PART  
C7D1 CD F7 C4          8700 CALL COMMA  ;",."  
C7D4 3E 41          8701 ST1: LD A,'A'  ;WRITE "A"  
C7D6 12          8702 LD (DE),A  
C7D7 13          8703 INC DE  
C7D8 C9          8704 RET       ;  
C7D9          8705 ;*****  
C7D9          8706 ; LD A,(BC) LD A,(DE) LD HL,(NN) LD A,(NN)  
C7D9          8707 ;*****  
C7D9 CD ED C4          8708 LDIND: CALL WRLD    ;WRITE "LD"  
C7DC F1          8709 POP AF      ;RECOVER OPCODE  
C7DD F5          8710 PUSH AF     ;SAVE  
C7DE FE 2A          8711 CP $2A      ;16 BIT?  
C7E9 28 21          8712 JR Z,LD16I  ;YES,JUMP  
C7E2 CD 04 C7          8713 CALL ST1     ;"A"  
C7E3 CD F7 C4          8714 CALL COMMA  ;",."  
C7EB F1          8715 POP AF      ;RECOVER OPCODE  
C7E9 FE 22          8716 LD1: CP $22      ;REGPR OR EXTENDED?  
C7EB EB          8717 EX DE,HL    ;WRITE "("  
C7EC 36 28          8718 LD (HL),(''  
C7EE 23          8719 INC HL  
C7EF EB          8720 EX DE,HL    ;  
C7F0 3F          8721 CCF      ;REVERSE RESULT OF COMPARE
```

REVAS subroutine version Ni.1 08-03-89

C7F1	04 77 C5	9722	CALL	NC,REGPR	; (RETURNS WITH NC)
C7F4	DC 32 C3	9723	CALL	C,LB1&A	; EXTENDED ADDRESS
C7F7	C3 F9 C3	9724	JP	NOTIXY	; CLOSE BRACKETS
C7FA		9725 ;			
C7FA		9726 ; 16-BIT INDIRECT STORE LD (NNNN),PP			
C7FA		9727 ;			
C7FA	CD E9 C7	9728 ST16I:	CALL	LD1	; DO INDIRECT BIT
C7FD	CD F7 C4	9729	CALL	COMMA	; ,
C800	C3 76 C5	9730	JP	REGPR-1	; GET OP & PR. REG PAIR
C803		9731 ;			
C803		9732 ; 16-BIT INDIRECT LOAD LD PP,(NNNN)			
C803		9733 ;			
C803	F1	9734 LB1&I:	POP	AF	; GET OPCODE
C804	CD 77 C5	9735	CALL	REGPR	; PRINT REG PR.
C807	CD F7 C4	9736	CALL	COMMA	; ,
C80A	A7	9737	AND	A	; CLEAR CARRY
C80B	18 DE	9738	JR	LD1+2	; DO INDIRECT BIT
C80D		9739 ;*****			
C80D		9740 ; ROTATE/SHIFT/BIT/SET/RESET			
C80D		9741 ; ND. IF INDEXED THEN OFFSET PRECEDES OPCODE			
C80D		9742 ;*****			
C800	3A 9C 9E	9743 CB1	LD	A,(HXYFLG)	
C816	A7	9744	AND	A	; INDEXED?
C811	F3	9745	PUSH	AF	; SAVE FLAG
C812	28 08	9746	JR	Z,NOTXY	; NO, SKIP
C814	11 34 9E	9747	LD	DE,BUFFER+32	; YES, WRITE..
C817	3E 06	9748	LD	A,6	; ...REG. FIRST.
C819	CD B4 C5	9749	CALL	SREG	
C81C	11 20 0E	9750	LD	DE,BUFFER+25	; RESET POINTER FOR MNEMONIC.
C81F	CD 08 C4	9751 NOTXY:	CALL	BYTE	; GET OPCODE
C822	F3	9752	PUSH	AF	; SAVE IT
C823	FE 49	9753	CP	\$49	; (\$49?)
C825	38 2C	9754	JR	C,ROTATE	; YES, JUMP
C827	21 4F CA	9755	LD	HL,BRSTAB-3	; LOAD POINTER
C82A	97	9756	RLCA		; SHIFT OPCODE DOWN
C82B	97	9757	RLCA		
C82C	E6 93	9758	AND	3	; ISOLATE ID
C82E	47	9759	LD	B,A	; MAKE 3,6,OR 9
C82F	97	9760	RLCA		
C830	88	9761	ADD	B	
C831	CD 18 C5	9762	CALL	FTADR	; FORM ADDRESS
C834	CD 38 C7	9763	CALL	COPY3S	; WRITE MNEMONIC
C837	F1	9764	POP	AF	; GET OPCODE
C838	F5	9765	PUSH	AF	; SAVE AGAIN
C839	9F	9766	RRCA		; PRINT BIT NUMBER
C83A	9F	9767	RRCA		
C83B	9F	9768	RRCA		
C83C	E6 07	9769	AND	7	; ISOLATE BIT ID
C83E	F6 38	9770	OR	\$38	; MAKE ASCII
C840	12	9771	LD	(DE),A	; WRITE IT
C841	13	9772	INC	DE	
C842	CD F7 C4	9773	CALL	COMMA	; ,
C843	C1	9774 TESTXY:	POP	BC	; RECOVER OPCODE
C846	F1	9775	POP	AF	; RECOVER HXY FLAG
C847	78	9776	LD	A,B	; LOAD OPCODE
C848	CA B4 C5	9777	JP	Z,SREG	; EXIT IF NOT HXY
C848	E6 97	9778	AND	17	; ISOLATE LO BITS OF OP
C84D	FE 96	9779	CP	6	; IS IT AN (HL) ONE?
C84F	C8	9780	RET	Z	; YES, RETURN
C850	C3 53 C9	9781	JP	NOTVAL	; NO, INVALID

REVAS subroutine version M1.1 88-03-09

C853	9F	9782	ROTATE:	RRCA	;SHIFT DOWN
C854	9F	9783		RRCA	
C855	C6 92	9784		ADD 2	;ROLL CODING ROUND
C857	E6 9E	9785		AND #E	;ISOLATE ID
C859	FE 9E	9786		CP #E	;IS IT #E?
C85B	CA 31 C9	9787		JP Z,NOTVAL	;YES,INVALID CODE
C85E	47	9788		LD B,A	;DO ID*3 AGAIN
C85F	9F	9789		RRCA	
C860	86	9790		ADD B	
C861	21 3D CA	9791		LD HL,ROTTAB	;LOAD BASE ADDRESS
C864	CD 18 C5	9792		CALL FTADR	;FORM ADDRESS
C867	CD 38 C7	9793		CALL COPY3S	;WRITE MNEMONIC
C86A	18 D9	9794		JR TESTXY	;EXIT WRITING REGISTER
C86C		9795	*****		
C86C		9796	*****		
C86C		9797	*****		
C86C	CB 57	9798	AUTO:	BIT 2,A	;TEST FOR VALIDITY
C86E	C2 33 C9	9799		JP NZ,NOTVAL	;JUMP IF NOT
C871	F3	9800		PUSH AF	;SAVE OPCODE
C872	E6 93	9801		AND 3	;ISOLATE OP ID
C874	#7	9802		RLCA	#2
C875	21 78 CA	9803		LD HL,OPTAB	;LOAD BASE ADDRESS
C878	CD 97 C5	9804		CALL RP2	;FORM ADDR. & COPY OVER
C87B	F1	9805		POP AF	;RECOVER OPCODE
C87C	F3	9806		PUSH AF	;SAVE AGAIN
C87D	3C	9807		INC A	;INC UP
C87E	E6 13	9808		AND \$13	;ISOLATE
C880	29 94	9809		JR NZ,++4	;SKIP IF NOT "OUT!"
C882	18	9810		DEC DE	;BACKSPACE POINTER
C883	CD 68 C5	9811		CALL COPY2	;MODIFY MNEMONIC
C886	F1	9812		POP AF	;RESET OPCODE
C887	21 7A CA	9813		LD HL,OPTAB+10	;LOAD BASE ADDRESS
C88A	9F	9814		RRCA	;SHIFT ID DOWN
C88B	9F	9815		RRCA	
C88C	E6 96	9816		AND 6	;ISOLATE IT
C88E	C3 97 C5	9817		JP RP2	;FINISH MNEMONIC
C891		9818	*****		
C891		9819	*****		
C891		9820	*****		
C891	21 E1 C9	9821	ADCSBC:	LD HL,ARTAB+9	;POINTER TO "SBC"
C894	79	9822		LD A,C	;LOAD OPCODE
C895	F3	9823		PUSH AF	;SAVE FOR "ADDHL"
C896	CB 5F	9824		BIT 3,A	;SBCT?
C898	28 93	9825		JR Z,++3	;YES,SKIP
C89A	21 DB C9	9826		LD HL,ARTAB+3	;POINT TO "ADC"
C89B	C3 6B C5	9827		JP ADDHL+3	;REST IN "ADDHL"
C8A0		9828	*****		
C8A0		9829	*****		
C8A0		9830	*****		
C8A0	3E FF	9831	IN:	LD A,\$FF	;IN A," ENTRY POINT
C8A2	FE 9E	9832	INRC:	CP #E	;TRY FOR (HL)?
C8A4	CA 33 C9	9833		JP Z,NOTVAL	;YES,INVALID
C8A7	21 49 CA	9834		LD HL,INH	;WRITE "IN"
C8AA	CD 38 C7	9835		CALL COPY3S	
C8AD	F3	9836		PUSH AF	;SAVE OP
C8AE	CD B4 C5	9837		CALL SREG	;WRITE REG.
C8B1	CD F7 C4	9838		CALL COMMA	;ADD ","
C8B4	F1	9839		POP AF	;RECOVER A
C8B5	EB	9840	PORT:	EX DE,HL	;OPEN BRACKETS
C8B6	36 28	9841		LD (HL),(''	

REVAS subroutine version M1.1 98-93-80

```

C8B8 23          9842    INC   HL
C8B9 3C          9843    INC   A           ;TEST OP.
C8BA 28 07        9844    JR    Z,INA       ;SKIP IF "IN A" (OR "OUT ..,A")
C8BC 36 43        9845    LD    (HL),`C'     ;SET "C"
C8BE 23          9846    INC   HL
C8BF EB          9847    EX    DE,HL       ;RESET ADDRESSES
C8C8 C3 F9 C5      9848    JP    NOTIXY     ;CLOSE BRACKETS
C8C3 EB          9849    INA: EX    DE,HL       ;GET & WRITE...
C8C4 CD F3 C5      9850    CALL  IMM          ;...PORT NUMBER
C8C7 18 F7        9851    JR    INA-3       ;CLOSE BRACKETS
C8C9
C8C9 ; EXTENDED DECODING
C8C9 ;*****9852 ;*****9853 ;*****9854 ;*****
C8C9 CD 88 C4        9855 EXTND: CALL  BYTE        ;GET OPCODE
C8CC FE C9          9856    CP    $CB         ;>=C8?
C8CE D2 53 C9        9857    JP    NC,NOTVAL   ;YES,NOT VALID
C8D1 FE 48          9858    CP    $48         ;<$48?
C8D3 DA 53 C9        9859    JP    C,NOTVAL   ;YES,NOT VALID
C8D6 FE A6          9860    CP    $A6         ;>=$A6?
C8D8 38 92          9861    JR    NC,AUTO     ;YES,AUTO INC/DEC
C8DA FE 88          9862    CP    $88         ;>=$88?
C8DC D2 53 C9        9863    JP    NC,NOTVAL   ;YES,NOT VALID
C8DF 4F          9864    LD    C,A         ;SAVE OPCODE
C8E9 E6 97          9865    AND   7           ;ISOLATE LQ BITS
C8E2 47          9866    LD    B,A         ;PUT IN B
C8E3 79          9867    LD    A,C         ;RELOAD OPCODE
C8E4 8F          9868    RRCA
C8E5 8F          9869    RRCA
C8E6 8F          9870    RRCA
C8E7 28 B9        9871    JR    Z,INRC      ;JUMP IF "IN"
C8E9 18 19        9872 NOTIN: DJNZ NOTOUT    ;TEST AGAIN
C8E9 ;*****9873 ;*****9874 ;*****9875 ;*****
C8EB ; OUT (XXX),A OUT (C),R
C8EB ;*****9876 ;*****9877 ;*****9878 ;*****9879 ;*****9880 ;*****9881 ;*****9882 ;*****9883 ;*****9884 ;*****9885 ;*****9886 ;*****9887 ;*****9888 ;*****9889 ;*****9890 ;*****9891 ;*****9892 ;*****9893 ;*****9894 ;*****9895 ;*****9896 ;*****9897 ;*****9898 ;*****9899 ;*****9900 ;*****9901 ;*****9902 ;*****9903 ;*****9904 ;*****9905 ;*****9906 ;*****9907 ;*****9908 ;*****9909 ;*****9910 ;*****9911 ;*****9912 ;*****9913 ;*****9914 ;*****9915 ;*****9916
C8EB 21          9876    DEF3  $21        ;SKIP THE NEXT 2 BYTES
C8EC 3E FF        9877 OUT: LD    A,$FF       ;"OUT -,A" ENTRY POINT
C8EE FE 2E        9878    CP    $2E         ;TRY FOR (HL)?
C8F0 CA 53 C9        9879    JP    Z,NOTVAL   ;YES,INVALID
C8F3 21 4C CA        9880    LD    HL,OUTM     ;POINT TO "OUT"
C8F6 CD 38 C7        9881    CALL  COPY3S    ;WRITE IT
C8F9 F3          9882    PUSH  AF         ;SAVE OP
C8FA CD 88 C9        9883    CALL  PORT       ;WRITE PORT (OR (C))
C8FD CD F7 C4        9884    CALL  CONNA      ;ADD ','
C8F9 F1          9885    POP   AF         ;RECOVER A
C901 C3 B4 C5        9886    JP    SREQ        ;WRITE REG.
C904
C904 ;*****9887 ;*****9888 ;*****9889 ;*****9890 ;*****9891 ;*****9892 ;*****9893 ;*****9894 ;*****9895 ;*****9896 ;*****9897 ;*****9898 ;*****9899 ;*****9900 ;*****9901 ;*****9902 ;*****9903 ;*****9904 ;*****9905 ;*****9906 ;*****9907 ;*****9908 ;*****9909 ;*****9910 ;*****9911 ;*****9912 ;*****9913 ;*****9914 ;*****9915 ;*****9916
C904 #5          9889 NOTOUT: DEC   B           ;TEST
C905 28 8A        9890    JR    Z,ADCSBC    ;JUMP IF "ADC" OR "SBC"
C907 18 14        9891    DJNZ NOTLD     ;TEST AGAIN
C909 ;*****9892 ;*****9893 ;*****9894 ;*****9895 ;*****9896 ;*****9897 ;*****9898 ;*****9899 ;*****9900 ;*****9901 ;*****9902 ;*****9903 ;*****9904 ;*****9905 ;*****9906 ;*****9907 ;*****9908 ;*****9909 ;*****9910 ;*****9911 ;*****9912 ;*****9913 ;*****9914 ;*****9915 ;*****9916
C909 ; LD PP,(SHNNN) LD (SHNNN),PP
C909 ;*****9900 ;*****9901 ;*****9902 ;*****9903 ;*****9904 ;*****9905 ;*****9906 ;*****9907 ;*****9908 ;*****9909 ;*****9910 ;*****9911 ;*****9912 ;*****9913 ;*****9914 ;*****9915 ;*****9916
C909 E6 9E        9895    AND   $E
C908 FE 9C        9896    CP    $C
C90D CA 53 C9        9897    JP    Z,NOTVAL
C910 CD ED C4        9898    CALL  WRLD       ;WRITE "LD"
C913 CB 59          9899    BIT   3,C         ;WHICH WAY?
C915 79          9900    LD    A,C         ;RELOAD OPCODE FOR LD/ST16
C916 C2 94 C8        9901    JP    NZ,LB16I+1  ;JUMP IF LOAD

```

REVAS subroutine version M1.1 #8-83-89

C919 F3	8982	PUSH AF	;AND PUT ON STACK
C91A C3 FA C7	8983	JP ST16I	;JUMP TO ST16I
C91B	8984	*****	
C91B	8985	; CLEAR UP THE REST OF EXETENBEB DECODING	
C91B	8986	*****	
C91B 79	8987	NOTLD: LD A,C	;RELOAD OPCODE
C91E 91 9C 88	8988	LD BC,12	;REMAINING OPS
C921 21 82 CA	8989	LD HL,REMEXT	;TABLE BASE
C924 ED 81	8990	CPIR	;CHECK TABLE
C926 28 28	8991	JR NZ,NOTVAL	;NOT THERE,ERROR
C928 79	8992	LD A,C	;GET REMAINING COUNT
C929 #7	8993	RLCA	;#2
C92A FE 88	8994	CP 8	;LD I OR R?
C92C 38 8E	8995	JR C,LDR1	;YES,SKIP
C92E #7	8996	RLCA	;#2 AGAIN
C92F 21 7E CA	8997	LD HL,EXTMNE-16	;BASE OF TABLE-16
C932 FE 1C	8998	CP 28	;IM?
C934 38 12	8999	JR C,INTMOD	;YES,SKIP
C936 CD 19 C5	9000	CALL FTABR	;INDEX INTO TABLE
C939 C3 97 C5	9001	JP COPY4	;COPY MNEMONIC OVER
C93C	9002	*****	
C93C	9003	LD A,R LD A,I LD R,A LD I,A	
C93C	9004	*****	
C93C CD ED C4	9005	LDR1: CALL WRD	;WRITE "LD"
C93F 21 AE CA	9006	LD HL,IRTAB	;LOAD BASE ADDRESS
C942 CD 19 C5	9007	CALL FTABR	;FORM ADDRESS
C943 C3 97 C5	9008	JP COPY3	;COPY OVER
C948	9009	*****	
C948	9010	; INTERRUPT MODE	
C948	9011	*****	
C948 CD 97 C5	9012	INTMOD: CALL RP2	;COPY OVER "IM"
C949 11 32 #E	9013	LD DE,BUFFER+39	;SET OPERAND ADDRESS
C94E ED A8	9014	LDI	;COPY OVER NODE #
C950 C9	9015	RET	;DONE
C951	9016	*****	
C951	9017	;INVALID CODE. WRITE 'DEFB' AND COPY CODE OVER.	
C951	9018	*****	
C951 E1	9019	NTVL: POP HL	;THROW AWAY TWO RETURNS
C952 E1	9020	POP HL	
C953 21 B7 CA	9021	LD HL,DEFB	;POINT TO "DEFB"
C956 11 2D 8E	9022	LD DE,BUFFER+29	;SET FOR MNEMONIC
C959 CB 97 C5	9023	CALL COPY4	;COPY IT OVER
C95C 13	9024	INC DE	;SPACE ON
C95B 21 1A 8E	9025	LD HL,BUFFER+4	;SET FOR OPCODES
C960 8E 15	9026	LD C,CR>8	;SET TO COUNT DEFBs
C962 3E 28	9027	LD A,' '	
C964 CD F9 C4	9028	NVLP: CALL POUND	;WRITE 'W'
C967 CB 98 C5	9029	CALL COPY2	;COPY OVER OP
C96A 23	9030	INC HL	;PAST SPACE
C96B BE	9031	CP (HL)	;IS ANOTHER THERE?
C96C 28 85 -	9032	JR Z,BRY	;NO,DONE
C96E CD F7 C4	9033	CALL COMMA	;WRITE ','
C971 18 F1	9034	JR NVLP	;COPY IT OVER
C973 21 88 88	9035	DNY: LD HL,0	;Clear any pointer
C976 22 12 8E	9036	LD (LABELP),HL	
C977 79	9037	LD A,C	;SEE HOW MANY ON LINE
C97A FE 8D	9038	CP CR	;DONE 4?
C97C C9	9039	RET NZ	;NO,RETURN
C97D 12	9040	LD (DE),A	;YES,SET CR SO NO RUBBISH
C97E C9	9041	RET	

REVAS subroutine version N1.1 98-93-88

```
C97F          9962 ;*****  
C97F          9963 ; TABLE AREA  
C97F          9964 ;*****  
C97F          9965 ;  
C97F          9966 ;JUMP TABLE FOR DECODE  
C97F          9967 ;32 ENTRIES  
C97F          9968 ;  
C97F #B C7    9969 TABLE: DEFU NOPETC  
C981 57 C6    9970 DEFU CJR  
C983 48 C5    9971 DEFU LD16  
C985 28 C6    9972 DEFU POP  
C987 C4 C7    9973 DEFU STIND  
C989 57 C6    9974 DEFU CJR  
C98B 33 C5    9975 DEFU INC  
C98D 55 C7    9976 DEFU JPETC  
C98F 33 C5    9977 DEFU INC  
C991 57 C6    9978 DEFU CJR  
C993 38 C5    9979 DEFU DEC  
C995 2D C6    9980 DEFU PUSH  
C997 9D C5    9981 DEFU LOADB  
C999 84 C6    9982 DEFU ARITH8  
C99B 86 C6    9983 DEFU ROTMIS  
C99D 93 C6    9984 DEFU RST  
C99F 47 C7    9985 DEFU EXAETC  
C9A1 57 C6    9986 DEFU CJR  
C9A3 68 C5    9987 DEFU ADDHL  
C9A5 98 C7    9988 DEFU RETETC  
C9A7 D9 C7    9989 DEFU LBIND  
C9A9 57 C6    9990 DEFU CJR  
C9AB 38 C5    9991 DEFU DEC  
C9AD 7C C7    9992 DEFU CBETC  
C9AF 33 C5    9993 DEFU INC  
C9B1 37 C6    9994 DEFU CJR  
C9B3 38 C5    9995 DEFU DEC  
C9B5 37 C6    9996 DEFU CALETC  
C9B7 9D C5    9997 DEFU LOADB  
C9B9 84 C6    9998 DEFU ARITH8  
C9B8 86 C6    9999 DEFU ROTMIS  
C9BD 93 C6    1000 DEFU RST  
C9BF          1001 ;*****  
C9BF 49 4E 43  1002 INCH: DEFB 'INC'  
C9C2 44 43 43  1003 DECM: DEFB 'DEC'  
C9C5 41 46 53  1004 DEFB 'AFS'  
C9C8 59 41 42 43 1005 RPRTAB: DEFB 'PABCDE'  
     44 45  
C9CE 48 4C 49 58 1006 HXYTAB: DEFB 'HIXIY'  
     49 59  
C9D4 48 41 4C 54 1007 HALTM: DEFB 'HALT'  
C9D8 41 44 44 41 1008 ARTAB: DEFB 'ADDADCSUBS3C'  
     44 43 53 55  
     42 53 42 43  
C9E4 41 4E 44 58 1009 DEFB 'ANDXOROR CP '  
     4F 32 4F 32  
     28 43 58 28  
C9F0 59 55 53 48 1010 PUSHN: DEFB 'PUSH'  
C9F4 58 4F 58 28 1011 POPN: DEFB 'POP '  
C9F8 52 45 54 28 1012 CJRTAB: DEFB 'RET JP CALL'  
     4A 58 28 28  
     43 41 4C 4C  
CA04 4E 5A 28 3A 1013 CCTAB: DEFB 'NZ ZNC CPOPE P N'
```

REVAS subroutine version M1.1 88-03-09

4E 43 29 43		
58 4F 58 45		
28 58 29 40		
CA14 52 4C 43 41	1914 RHTAB:	DEFB 'RLCARRCARLA RRA'
52 52 43 41		
32 4C 41 29		
32 52 41 28		
CA24 44 41 41 28	1915	DEFB 'DAA CPL SCF CCF'
43 58 4C 28		
33 43 46 28		
43 43 46 28		
CA34 52 53 54	1916 RSTM:	DEFB 'RST'
CA37 4E 4F 58	1917 NOPN:	DEFB 'NOP'
CA3A 44 4A 4E 5A	1918 DJNZH:	DEFB 'DJNZ'
CA3E 41 46 2C 41	1919 EXAFH:	DEFB 'AF,AF///'
46 27		
CA44 28 53 58 29	1920 BSPBM:	DEFB '(SP),'
2C		
CA49 49 4E 28	1921 INH:	DEFB 'IN'
CA4C 4F 53 54	1922 OUTH:	DEFB 'OUT'
CA4F 53 54 2C	1923 SPM:	DEFB 'SP,'
CA52 42 49, 54 52	1924 BRSTAB:	DEFB 'BITRESET'
45 53 53 43		
54		
CA5B 53 52 4C 52	1925 ROTTAB:	DEFB 'SRLRLCRRCRL RR SLASRA'
4C 43 52 52		
43 52 4C 28		
52 52 29 33		
4C 41 53 52		
41		
CA70 4C 44 43 58	1926 OPTAB:	DEFB 'LDCPINOTUTI D IRDR'
49 4E 4F 54		
55 54 49 28		
44 28 49 52		
44 52		
CA82 44 45 4D 67	1927 REMEXT:	DEFB \$44,\$45,\$4D,\$67,\$6F,\$46
6F 46		
CA88 54 5E 57 4F	1928	DEFB \$56,\$5E,\$57,\$4F,\$5F,\$47
5F 47		
CA8E 49 4D 32 28	1929 EXTNHE:	DEFB 'IM2 IM1 IM0 RLD'
49 4D 31 28		
49 4D 38 28		
52 4C 44 28		
CA9E 52 52 44 28	1930	DEFB 'RRD RETIRETNNEG'
52 45 54 49		
52 43 54 4E		
4E 43 47 28		
CAAE 49 2C 41 2C	1931 IRTAB:	DEFB 'I,A,R,A,I'
52 2C 41 2C		
49		
CAB0	1932 ACNMMAS:	EQU IRTAB+2
CAB7 44 45 46 42	1933 DEFB:	DEFB 'DEFB'
CABB	1934 *****	*****
CABB	1935 REVASC:	EQU *
	1936	END

CROSS REFERENCE

ACONMA	CAB0	0424
ADCSBC	C891	0890
ADDHL	C568	0827 0987
ADDR	0E94	0100 0129 0180 0182 0393
ARG3	0C18	0825 0938
ARES	0668	0821
ARITH8	C604	0262 0982 0998
ARTAB	C9D8	0311 0420 0821 0826
AUTO	C86C	0861
BORN	C6B8	0539 0548
BSTTAB	CA52	0735
BSPBH	CA44	0632
BUFFER	0E14	0117 0123 0142 0218 0523 0539 0582 0673 0747 0759 0933 0942 0945
BYTE	C4B8	0119 0299 0301 0380 0400 0452 0533 0541 0553 0588 0751 0855
CALETC	C637	0996
CB	C880	0633
CBETC	C77C	0992
CCODES	C67C	0476 0586
CCTAB	CA94	0488
CJR	C457	0446 0626 0667 0970 0974 0978 0986 0998 0994
CJRTAB	C9F8	0466 0671
COMMA	C4F7	0298 0315 0353 0480 0587 0668 0700 0714 0729 0736 0773 0838 0884 0953
COPY2	C59B	0331 0340 0426 0428 0659 0661 0672 0811 0949
COPY3	C369	0600 0686 0928
COPY3S	C738	0284 0312 0423 0503 0763 0793 0835 0881
COPY4	C567	0407 0436 0468 0498 0508 0921 0943
COPY5	C595	0633
COPY6	C393	0628
CR	0890	0936 0941 0946 0958
DEC	CS38	0979 0991 0995
DECM	C9C2	0283
DECODE	C315	0111 0458
DEFB	CAB7	0941
DI	C779	0629
DJNZ	C73E	0573
DJNZH	CA3A	0607
DNV	C973	0952
DONEH	C6D1	0532
EADDR	0E98	0102 0104
EI	C774	0633
ESC	081B	0946
EXAETC	C747	0983
EXAFH	CA3E	0619
EXTNNE	CABE	0917
EXTND	C8C9	0444
EXX	C7BD	0668
FLUSH	C6E9	0514 0519
FTADR	C319	0269 0339 0365 0421 0467 0497 0762 0792 0920 0927
HALT	C5FE	0346
HALTH	C9D4	0496
HEX1	C4A9	0162
HEX2	C4A2	0151 0198 0482
HEX4	C49D	0131 0307
HXYFLG	0E8C	0112 0128 0336 0337 0372 0373 0451 0743
HXYTAB	C9CE	0371
INH	C5F3	0356 0858
IN	C8A0	0652

CROSS REFERENCE

INA	CBC3	9844 9851
INC	C533	9975 9977 9993
INCH	C9BF	9281
INITB	C46E	9199 9552
INH	CA49	9834
INRC	C842	9871
INTMOD	C948	9919
IRTAB	CAA8	9926 1932
JPETC	C753	9976
JR	C712	9616
LBB	C5AF	9429
LABELA	SE10	9365
LABELP	SE12	9127 9364 9956
LDI	C7E9	9874 9699 9728 9738
LD16	C548	9971
LD16A	C552	9481 9723
LD16B	C558	9593
LD16I	C883	9712 9991
LDIND	C7D9	9989
LDR1	C93C	9915
LISP	C7AF	9679
LINEA	SE8E	9139
LOAD8	C590	9260 9981 9997
NADDR	SE86	9181 9185 9183 9186 9558
MEM	C5C4	9363
MORENS	C6C3	9536
MS1	CACD	9539
NEXTL	C449	9119
NOP	C733	9574
NOPETC	C788	9969
NOPM	CA37	9599
NOTIN	CBE9	
NOTIXY	C5F0	9378 9383 9724 9848
NOTLB	C91D	9891
NOTOUT	C944	9872
NOTSP	C38D	9328
NOTVAL	C953	9116 9543 9547 9555 9781 9799 9833 9857 9859 9863 9879 9897 9911
NOTXY	C81F	9746
NTVL	C951	9787
NUMARG	9C88	9917
NVLP	C964	9954
OPCADR	SE8A	9134 9187 9192
OPTAB	CA78	9883 9813
OUT	C8EC	9627
OUTH	CA4C	9889
PATCH	SE88	9913 9916 9119 9531
PHEX2	C5F8	9392 9596
PLUS	C5E8	9384
PDP	C628	9772
POPN	C9F4	9433
PORT	C895	9883
POUND	C4FB	9366 9461 9948
PRINT	C428	9915 9937
PRTABL	C6FC	9531 9344
PRTBI	C788	9357
PUSH	C62D	9988
PUSHM	C9F8	9435
REGPR	C577	9287 9297 9314 9438 9635 9682 9722 9738 9735
RENEXT	CAB2	9989

CROSS REFERENCE

RETETC	C798	0988
RETURN	0958	0949
REVAS	C43E	0929
REVASC	CAB8	0919
RIN	0908	0945
RNTAB	CA14	0946
ROTATE	C853	0754
ROTHIS	C686	0983 0999
ROTTAB	CA48	0791
RP2	C597	0335 0375 0489 0804 0817 0932
RPRTAB	C9C8	0327 0364 0658
RST	C693	0984 1000
RSTFLB	0E83	0914 0987
RSTM	CA34	0302
SAVE	C43E	0999
SCAL	0918	0929 0948
SPH	CA4F	0679
SREG	C584	0291 0332 0749 0777 0837 0884
ST1	C7D4	0713
ST16I	C7FA	0697 0993
STIND	C7C4	0973
TABLE	C97F	0268
TESTXY	C845	0794
UCD	C727	0384 0610
UN0	C6D6	0546
UN1	C6E4	0545
UNCMD	C679	0474
UNPRN	C6D9	0521
UREX	C4E3	0618 0631 0657 0684
WRLD	C4ED	0295 0348 0678 0693 0798 0898 0925
WRLD#	C4F3	0219

Interactive control program for REVAS

```
C499      9992 ;
C499      9993 ; EQUATES TO MAS-SYS ROUTINES
C499      9994 ;
      9995 RIN:    EQU   8
      9996 SCAL:   EQU   $18
      9997 PRS:    EQU   $28
      9998 ROUT:   EQU   $30
      9999 RDEL:   EQU   $38
      9910 RETURN: EQU   $58
      9911 ARGS:   EQU   $60
      9912 POLLK:  EQU   $62
      9913 INLINE: EQU   $63
      9914 NUM:    EQU   $64
      9915 CRLF:   EQU   $6A
      9916 ERRM:   EQU   $6B
      9917 SRLX:   EQU   $6F
      9918 RLIN:   EQU   $79
      9919 BLINK:  EQU   $7B
      9C93     9920 ARGN:   EQU   $C0B
      9921 NUMV:   EQU   $C21
C499      9922 ;
C499      9923 ; EQUATES TO REVAS ROUTINES
C499      9924 ;
      C43E     9923 REVAS:  EQU   **+$3E
      C46E     9924 INITB:  EQU   **+$4E
      C49D     9927 HEX4:   EQU   **+$90
      C498     9928 BYTE:   EQU   **+$92
      C933     9929 NOTVAL: EQU   **+$33
C499      9930 ;
      9931 LF:    EQU   $A
      9932 CR:    EQU   $D
      9933 ESC:   EQU   $1B
C499      9934 ;*****;
C499      9935 ; VARIABLES AND BUFFERS
C499      9936 ;*****;
      CABB     9937 SAVE:  EQU   **+$4B9
      C499      9938 ORG   9E99
      9E99  9939 PATCH: DEFS  3          ;RAM PATCH TO O/P ROUTINE
      9E93  9940 RSTFLG: DEFS  1          ;RST HANDLING FLAG
      9E94  9941 ADDR:  DEFS  2
      9E96  9942 HADDR: DEFS  2
      9E98  9943 EADDR: DEFS  2
      9E9A  9944 OPCADR: DEFS  2
      9E9C  9945 HXYFLG: DEFS  2
      9E9E  9946 LINEA: DEFS  2          ;ADDR. LINE REF. LABEL
      9E10  9947 LABELA: DEFS  2         ;LABEL VALUE
      9E12  9948 LABELP: DEFS  2         ;POINTER TO POS IN BUFFER
      9E14  9949 BUFFER: DEFS  48        ;OUTPUT BUFFER
      9E44  9950 FLAG:   DEFS  1          ;FLAG BYTE
      9E45  9951 STEP:   DEFS  1          ;SINGLE STEP FLAG
      9E46  9952 START:  DEFS  2          ;START OF DATA ON STACK
      9E48  9953 FIRST:  DEFS  2          ;RANGE
      9E4A  9954 LAST:   DEFS  2
      9E4C  9955 POINT:  DEFS  2
      9E4E  9956 SYMST:  DEFS  2          ;SYMBOL TABLE START
      9E50  9957 SYMEND: DEFS  2         ;SYMBOL TABLE END
      9E52  9958 PSYN:   DEFS  2          ;PRESENT POSITION IN TABLE
      9E54  9959 OFFSET: DEFS  2
      9E56  9960 ZEAPP:  DEFS  2          ;POINTER INTO ZEAP FILE
      9E58  9961 SEQU:   DEFS  36
```

Interactive control program for REVAS

0E7C 0091	0062 LC:	DEFS 1	;LINE COUNT
0E7D 0091	0063 GAP:	DEFS 1	
0E7E 0092	0064 LINES:	DEFS 2	
0E80 0092	0065 LINENO:	DEFS 2	;ZEEP LINE NUMBER
0E82 0092	0066 ZLABEL:	DEFS 2	;ZEEP LABEL COUNT*2
0E84 0098	0067 HEAD:	DEFS 8	
0E8C 0096	0068 PAGE:	DEFS 6	
0E92 0031	0069 TITLE:	DEFS 49	;ROOM FOR TITLE

Interactive control program for REVAS

```

9EC3      9871 ;
9EC3      9872 ;     A CONTROL PROGRAM FOR REVAS
9EC3      9873 ;
9EC3      9874 ;     WRITTEN BY DAVID PARKINSON
9EC3      9875 ;
9EC3      9876 ;     VERSION N1.1  98-03-09
9EC3      9877 ;
9EC3      9878     ORG  SAVE
CABB      9879 ;*****INITIALISATION FOR DATA AREA
CABB      9880 ;     INITIALISATION FOR DATA AREA
CABB      9881 ;*****INITIALISATION FOR DATA AREA
CACB 11 88 8E 9882 REVASC: LD DE,LINENO
CACB 21 E5 CF 9883 LD HL,RAMLD
CAC1 #1 12 #8 9884 LD BC,MOVING
CAC4 ED 89 9885 LDIR
CAC6      9886 ;*****FIRST SET UP ALL PROGRAM CONTROL VARIABLES
CAC6      9887 ;     FIRST SET UP ALL PROGRAM CONTROL VARIABLES
CAC6      9888 ;*****FIRST SET UP ALL PROGRAM CONTROL VARIABLES
CAC6 EF 9889 OPTIONS: RST PRS ;ASK FOR OPTIONS
CAC7 9D 4F 78 74 9890 DEFB CR,'Options? '
69 6F 6E 73
3F 29 28
CAC2 53 54 5A 58 9891 OTAB: DEFB 'STIXLPDR' ;OPTION TABLE
4C 58 44 52
CADA 53 29 2D 96 9892 DEFB 'U)-',0
CADE 11 98 96 9893 LD DE,B ;CLEAR D & E
CAE1 3E 91 9894 LD A,1 ;SET LSB OF A
CAE3 #1 9895 DEFB 1 ;SKIP THE NEXT TWO BYTES
CAE4 D6 53 9896 FOUND: SUB 'U' ;IS IT U?
CAE6 29 93 9897 JR NZ,*+3 ;NO,LEAVE ALONE
CAE8 32 93 8E 9898 LD (RSTFLG),A ;SET FLAG
CAEB 7B 9899 LD A,E ;SET APP. BIT IN FLAG
CAEC B2 9100 OR D
CAED 57 #101 LD D,A
CAEE CF 9102 RST RIN ;READ A CHARACTER
CAEF F7 9103 RST ROUT ;ECHO THE CHARACTER
CAF0 37 9104 SCF ;SET FLAG
CAF1 1E 98 9105 LD E,0 ;CLEAR E
CAF3 21 D2 CA 9106 LD HL,OTAB ;LOOK UP OPTION
CAF6 CB 13 9107 LUOP: RL E ;ROTATE E
CAF8 ED A1 9108 CPI ;CHECK AGAINST TABLE
CAFA 28 E8 9109 JR Z,FOUND ;JUMP IF THERE
CAF0 30 F8 9110 JR NC,LUOP ;TRY AGAIN IF FLAG NOT THRU'
CAFE FE 9D 9111 CP CR ;SEND?
CB00 28 C4 9112 JR NZ,OPTION ;UNRECOGNISED,TRY AGAIN
CB02 7A 9113 LD A,D ;GET OPTIONS
CB03 CB 57 9114 BIT 2,A ;ZAP FILE?
CB05 28 92 9115 JR Z,*+2 ;NO,SKIP
CB07 F6 11 9116 OR $11 ;YES,SET LABEL & SOURCE
CB09 32 44 8E 9117 LD (FLAG),A
CB0C F3 9118 PUSH AF ;SAVE FLAG
CB0D
9119 ;
CB0D 9120 ;OPTIONS SET IN FLAG
CB0D 9121 ;
CB0D 9122 ; BIT IF SET
CB0D 9123 ;"S" 0 - Source file.(No addr/ops or format)
CB0D 9124 ;"T" 1 - Tape output
CB0D 9125 ;"Z" 2 - ZAP source file to memory
CB0D 9126 ;"X" 3 - Xreference table
CB0D 9127 ;"L" 4 - Labelled listing

```

Interactive control program for REVAS

CB9B	\$128 ;"P" 5	-	Paginated listing
CB9B	\$129 ;"B" 6	-	Insert end-of-line delay
CB9D	\$130 ;"R" 7	-	Range limitation on listing
CB9E	\$131 ;		
C90D E6 1C	\$132 AND \$1C		;XREF,LABELS OR ZEAP?
C90F 28 38	\$133 JR Z,NLABS		;NO,SKIP
CB11	\$134 ;		
CB11	\$135 ; GET SYMBOL TABLE MEMORY BOUNDS		
CB11	\$136 ;		
CB11 F1	\$137 GETSTA: POP AF		;RESET FLAG IN A
CB12 F5	\$138 PUSH AF		
CB13 CB 57	\$139 BIT. 2,A		;ZEAP FILE?
CB15 28 98	\$140 JR Z,ST		;NO,DIFFERENT MESSAGE
CB17 EF	\$141 RST PRS		
C918 5A 45 41 58	\$142 DEFB 'ZEAP f1',0		
28 66 69 00			
CB20 1B 0C	\$143 JR STB		
CB22 EF	\$144 ST:	RST PRS	
CB23 53 79 60 62	\$145 DEFB 'Symbol tab',0		
6F 6C 28 74			
61 62 00			
CB2E EF	\$146 STB:	RST PRS	
C92F 6C 65 29 61	\$147 DEFB 'Is area?',CR,0		
72 65 61 3F			
68 00			
CB39 CB FE CC	\$148 CALL GETTWO		;GET BOUNDS
CB3C 38 03	\$149 JR C,GETSTA		;REPEAT ON ERROR
CB3E EB 43 4E 9E	\$150 LD (SYHST),BC		
CB42 ED 53 39 9E	\$151 LD (SYHEND),DE		
CB46 3E FF	\$152 LD A,IFF		;SET EOT MARKER
CB48 12	\$153 LD (9E),A		
CB49	\$154 ;		
CB49	\$155 ; LOAD TITLE IF PAGINATED & SET PAGE NO.		
CB49	\$156 ;		
CB49 F1	\$157 NLABS: POP AF		;RECOVER FLAG
CB4A CB 6F	\$158 BIT 5,A		
CB4C 28 38	\$159 JR Z,NOTITL		;SKIP IF NONE
CB4E EF	\$160 ASKT: RST PRS		;ELSE ASK FOR IT
C94F 54 69 74 6C	\$161 DEFB 'Title?',CR,0		
65 3F 00 00			
CB57 BF	\$162 RST SCAL		;READ IT
CB58 63	\$163 DEFB INLINE		
CB59 21 3C 05	\$164 LB HL,\$53C		;DEFAULT VALUES FOR LISTING
CB5C 1A	\$165 LB A,(DE)		;GET FIRST CHARACTER
CB5D FE 3D	\$166 CP "z"		;REDEFINE?
CB5F 28 00	\$167 JR NZ,SETLPP		;NO,SKIP
CB61 13	\$168 INC BE		;YES,SPACE PAST "="
CB62 BF	\$169 RST SCAL		
CB63 64	\$170 DEFB NUM		;READ VALUE
CB64 38 E8	\$171 JR C,ASKT		;REPEAT ON ERROR
CB66 2A 21 0C	\$172 LB HL,(NUMV)		;SET NEW VALUE
CB67 22 7E 9E	\$173 SETLPP: LB (LINES),HL		;SET VALUES
CB6C 21 91 01	\$174 LD HL,\$191		;SET FOR FIRST ONE
CB6F 22 7C 9E	\$175 LD (LC),HL		
CB72 EB	\$176 EX BE,HL		;SOURCE TO HL
CB73 11 92 9E	\$177 LD BE,TITLE		;BUFFER
CB76 AF	\$178 XOR A		;CLEAR A
CB77 EB A9	\$179 LBI		;COPY OVER UNTIL NULL
CB79 BE	\$180 CP (HL)		
C97A 28 FB	\$181 JR NZ,=S		

Interactive control program for REVAS

CB7C E8	#182	EX DE,HL	;END TO HL
CB7D 3E 29	#183	LD A,' '	;NOW GO BACK DOWN BUFFER...
CB7F 2B	#184	DEC HL	
C886 BE	#185	CP (HL)	;..LOOKING FOR FIRST...
C881 2B FC	#186	JR Z,*-4	;....NON SPACE.
C883 23	#187	INC HL	;CORRECT HL
C884 36 9D	#188	LD (HL),CR	;SET CR AT END
C886	#189 ;		
C886	#190 ;	NOW FIND OUT WHAT TO RUN ON	
C886	#191 ;		
C886 EF	#192 NOTITL: RST PRS		;GET ADDRESSES FOR REVAS
CB87 37 68 61 74	#193 DEFB 'What on?',CR,0		
26 6F 6E 3F			
6D 98			
CB91 DF	#194 RST SCAL		;READ LINE
CB92 63	#195 DEFB INLINE		
CB93 DF	#196 RST SCAL		;GET ADDRESSES TO ARGS
CB94 79	#197 DEFB RLIN		
CB95 38 EF	#198 JR C,NOTITL		;TRY AGAIN IF NON-HEX
CB97 3A 9B 9C	#199 LD A,(ARGN)		;NO. OF ARGS
CB9A FE 82	#200 CP 2		;CHECK >=2 ENTERED
C89C 38 E8	#201 JR C,NOTITL		;NO,REPEAT QUESTION
CB9E DF	#202 RST SCAL		;GET ARGUEMENTS
CB9F 68	#203 DEFB ARGS		
C8A0 13	#204 INC DE		;SAVE END ADDRESS
C8A1 ED 53 14 8E	#205 LD (BUFFER),DE		
C8A5 34	#206 LD B,H		;HL->DE
C8A6 5D	#207 LD E,L		
CB87 2B	#208 DEC HL		;ADJUST
CBAB ED 73 46 8E	#209 LD (START),SP		;SET POINTER
CBAC E3	#210 PUSH HL		;SAVE START
CBAD 23	#211 INC HL		;CORRECT
CBAE 28 82	#212 JR Z,*+2		;DON'T LOAD THIRD IF ONLY 2
C8B0 68	#213 LD H,B		;OVERWRITE WITH 3RD
C8B1 69	#214 LD L,C		
C8B2 ED 52	#215 SBC HL,DE		;COMPUTE OFFSET
C8B4 22 54 8E	#216 LD (OFFSET),HL		;SAVE
C8B7	#217 ;		
C8B7	#218 ;	CHECK FOR RESTRICTED RANGE TO PRINT	
C8B7	#219 ;		
C8B7 #1 98 99	#220 LD BC,8		;LOAD DEFAULT VALUES
C8BA 11 FF FF	#221 LD DE,-1		
C8BD 21 44 8E	#222 LD HL,FLAG		;POINT TO FLAG BYTE
CBC9 CB 7E	#223 BIT 7,(HL)		;RANGE WANTED?
CBC2 28 18	#224 JR Z,DA		;NONE,SKIP
CBC4 CB 9E	#225 RES 7,(HL)		;CLEAR FOR PASS 1 FLAG
CBC6 EF	#226 RNGRO: RST PRS		;ASK FOR RANGE
CBC7 4C 69 73 74	#227 DEFB 'Listing range?',CR,0		
69 6E 67 28			
72 61 6E 67			
65 3F 9D 99			
CB07 CD FE CC	#228 CALL GETTWO		;GET IT
CB0A 38 EA	#229 JR C,RNGRO		;REPEAT ON ERROR
CB0C ED 43 48 8E	#230 DAS LD (FIRST),BC		;SET VALUES
CDE9 ED 53 4A 8E	#231 LD (LAST),DE		
CDE4	#232 ;		
CDE4	#233 ;	ASK FOR ANY DATA AREAS	
CDE4	#234 ;		
CDE4 EF	#235 DAREA: RST PRS		;ASK FOR DATA AREAS
CBE3 44 41 54 41	#236 DEFB 'DATA areas?',CR,0		

Interactive control program for REVAS

29 61 72 65				
61 73 3F 8D				
99				
CBF2 AF	#237	XOR A	;CLEAR COUNT	
C3F3 F5	#238	PUSH AF	;SAVE COUNT	
C3F4 CD FE CC	#239 DAREA8:	CALL GETTWO	;GET NEXT PAIR	
C3F7 38 27	#240	JR C,DAREA1	;SKIP ON ERROR	
C3F9 68	#241	LD H,B	;ADJUST VALUES BY OFFSET	
C3FA 69	#242	LD L,C		
CBFB ED 4B 54 8E	#243	LD BC,(OFFSET)		
CBFF C5	#244	PUSH BC		
CC98 ED 42	#245	SBC HL,BC		
CC92 E3	#246	EX (SP),HL		
CC93 C1	#247	POP BC		
CC94 E8	#248	EX DE,HL		
CC95 B7	#249	DR A	;CLEAR CARRY	
CC96 ED 52	#250	SBC HL,DE		
CC98 EB	#251	EX DE,HL		
CC99 F1	#252	POP AF	;RECOVER COUNT	
CC9A E1	#253	POP HL	;GET PREVIOUS	
CC9B E3	#254	PUSH HL		
CC9C F3	#255	PUSH AF	;RESET	
CC9D 37	#256	SCF	;COMPARE THEM	
CC9E 23	#257	INC HL		
CC9F ED 42	#258	SBC HL,BC		
CC11 38 24	#259	JR NC,DAREA2	;JUMP ON ERROR	
CC13 2A 14 8E	#260	LD HL,(BUFFER)	;CHECK END>=2ND	
CC16 ED 52	#261	SBC HL,DE		
CC18 38 10	#262	JR C,DAREA2		
CC1A F1	#263	POP AF	;REMOVE COUNT	
CC1B C5	#264	PUSH BC	;OK,SO SAVE	
CC1C D3	#265	PUSH DE		
CC1D 3C	#266	INC A	;BUMP COUNT	
CC1E 18 D3	#267	JR DAREA8-1	;GO GET NEXT	
CC20 A7	#268 DAREA1:	AND A	;TEST A	
CC21 28 10	#269	JR Z,DAREA3	;CONTINUE IF END	
CC23 FE 2D	#270	CP '-'	;DELETE LAST?	
CC25 28 19	#271	JR NZ,DAREA2	;NO,ERROR	
CC27 F1	#272	POP AF	;RECOVER COUNT	
CC28 A7	#273	AND A	;ZERO?	
CC29 28 39	#274	JR Z,DAREA	;YES,START AGAIN	
CC2B E1	#275	POP HL	;NO,DELETE LAST	
CC2C E1	#276	POP HL		
CC2D 3D	#277	DEC A		
CC2E F3	#278	PUSH AF	;RESET COUNT	
CC2F EF	#279	RST PRS	;MOVE CURSOR BACK	
CC30 13 1B 13 1B	#280	DEFB \$13,ESC,\$13,ESC,\$		
99				
CC35 18 BD	#281	JR DAREA8	;LOOP	
CC37 DF	#282 DAREA2:	RST SCAL	;PRINT 'Error'	
CC38 4B	#283	DEFB ERRM		
CC39 EF	#284	RST PRS	;MOVE CURSOR BACK	
CC3A 13 13 17 88	#285	DEFB \$13,\$13,\$17,\$		
CC3E 10 B4	#286	JR DAREA8	;TRY AGAIN	
CC40	#287 ;			
CC41 F1	#288 DAREA3:	POP AF	;REMOVE COUNT	
CC41 2A 14 8E	#289	LD HL,(BUFFER)	;GET END ADDRESS	
CC44 E3	#290	PUSH HL	;ADD TO LIST	
CC45 21 FF FF	#291	LD HL,-1	;SET END MARKER	
CC48 E5	#292	PUSH HL		

Interactive control program for REVAS

```

CC49          9293 ;
CC49          9294 ; GET SINGLE STEP/RUN CHARACTER
CC49          9295 ;
CC49  EF      9296     RST    PRS
CC4A 47 6F 3F 98 9297     DEF3  'Go?',#
CC4E  DF      9298     RST    SCAL
CC4F  78      9299     DEFB  BLINK
CC50  32 45 9E 9300     LD    (STEP),A    ;SAVE IN STEP FLAG
CC53  DF      9301     RST    SCAL    ;NEW LINE
CC54  6A      9302     DEFB  CRLF
CC55          9303 ;*****;
CC55          9304 ; EVERYTHING NOW SET UP, LET'S GO
CC55          9305 ;*****;
CC55  3A 44 9E 9306     LD    A,(FLAG)  ;SEE IF PASS1 NEEDED
CC58  E6 18      9307     AND   $18     ;XREF OR LABELS?
CC5A  21 88 CD  9308     LD    HL,PASS1 ;(SET O/P DRIVER ADDR)
CC5D  C4 37 CD  9309     CALL  NZ,PASS  ;YES,DO FIRST PASS
CC60  21 44 9E  9310     LD    HL,FLAG  ;POINT TO FLAG
CC63  C8 FE      9311     SET   7,(HL)   ;SET PASS2 FLAG
CC65  C8 56      9312     BIT   2,(HL)   ;ZEEP FILE?
CC67  28 23      9313     JR    Z,ALLING ;NO,SKIP
CC69          9314 ;
CC69          9315 ; FOR ZEEP FILE - SET ADDRESSES IN ZEEP
CC69          9316 ; WORKSPACE AND INITIALISE FILE
CC69          9317 ;
CC69  2A 58 9E  9318     LD    HL,(SYMEND) ;SET END FILE AREA
CC6C  22 92 0F  9319     LD    ($F92),HL
CC6F  2A 4E 9E  9320     LD    HL,(SYNST)  ;SET START OF AREA
CC72  22 88 9F  9321     LD    ($F88),HL
CC73  ED 5B 52 0E 9322     LD    DE,(PSYM)  ;GET CURRENT END OF TABLE
CC77  ED 53 4E 9E 9323     LD    (SYNST),DE ;SET IN SYNST
CC7D  ED 5B 82 0E 9324     LD    DE,(ZLABEL) ;SET LABEL COUNT
CC81  23      9325     INC   HL        ;...AT START OF FILE.
CC82  23      9326     INC   HL
CC83  73      9327     LD    (HL),E
CC84  23      9328     INC   HL
CC85  72      9329     LD    (HL),D
CC86  23      9330     INC   HL
CC87  36 88      9331     LD    (HL),#    ;SET INITIAL NULL
CC89  22 56 9E  9332     LD    ($EAPP),HL ;SET FILE POINTER
CC8C          9333 ;
CC8C  21 25 CE  9334 ALLING: LD    HL,PASS2 ;(SET O/P DRIVER ADDR.)
CCBF  CD 37 CD  9335     CALL  PASS    ;GO DO IT
CC92  21 44 9E  9336     LD    HL,FLAG  ;POINT TO FLAG
CC95  C8 9E      9337     RES   7,(HL)   ;CLEAR PASS 2 FLAG
CC97  CB 66      9338     BIT   4,(HL)   ;LABELS?
CC99  E3      9339     PUSH  HL        ;(SAVE HL)
CC9A  C4 92 CF  9340     CALL  NZ,LABEL ;YES,PRINT REMAINING.
CC9D  E1      9341     POP   HL        ;(RESET HL)
CC9E  CB 5E      9342     BIT   3,(HL)
CCA6  29 92      9343     JR    NZ,*+2  ;YES,DO IT
CCA2  DF      9344 EXIT:  RST    SCAL    ;NO,RETURN
CCA3  5B      9345     DEFB  RETURN
CCA4          9346 ;
CCA4          9347 ; PRINT THE CROSS REFERENCE TABLE
CCA4          9348 ;
CCA4  7E      9349     LD    A,(HL)   ;GET FLAG
CCA5  21 7D 9E  9350     LD    HL,GAP  ;POINT TO GAP
CCA8  35      9351     DEC   (HL)    ;REDUCE BY ONE
CCA9  28      9352     DEC   HL        ;POINT TO LC

```

Interactive control program for REVAS

CCAA E5	0353	PUSH HL	;SAVE HL)
CCAB CB 6F	0354	BIT 5,A	;CHECK IF PAGED
CCAB C4 88 CF	0355	CALL NZ,EJECT	;NEW PAGE IF PAGED
CCB0 E1	0356	POP HL	;RECOVER POINTER TO LC
CCB1 34	0357	INC (HL)	;INCREASE BY ONE
CCB2 21 FF FF	0358	LB HL,FFFF	;RESET RANGE HI LIMIT
CCB3 22 4A 8E	0359	LB (LAST),HL	
CCB8 FB 2A 58 8E	0360	LB IY,(SYMEND)	;SET POINTER TO TABLE
CCBC FB 7E 88	0361 XREF1:	LB A,(IY)	;CHECK FOR EOT
CCBF 3C	0362	INC A	;THERE?
CCC0 28 E8	0363	JR Z,EXIT	;YES,FINISHED
CCC2 FD 2B	0364	BEC IY	;SPACE ON
CCC4 4F	0365	LB C,A	;PICK UP COUNT
CCC5 94 87	0366	LB B,9	;SET 9 ENTRIES MAX ON LINE
CCC7 11 14 8E	0367	LB BE,BUFFER	;POINT TO BUFFER
CCCA FB 6E 88	0368 XREF2:	LB L,(IY)	;LOAD REFERENCE
CCC9 FD 2B	0369	DEC IY	
CCCF FD 66 88	0370	LB H,(IY)	
CCB2 FB 2B	0371	DEC IY	
CCD4 CB 9B C4	0372	CALL HEX4	;WRITE TO BUFFER
CCD7 3E 2B	0373	LD A,929	;SPACE
CCD9 12	0374	LD (DE),A	
CCDA 13	0375	INC BE	
CCDB 80	0376	DEC C	;MORE?
CCDC 28 82	0377	JR Z,++2	;NO,DONE ENTRY
CCDE 18 EA	0378	BINZ XREF2	;LOOP IF MORE ON LINE
CCE0 C5	0379	PUSH BC	;SAVE COUNT
CCE1 E9	0380	EX DE,HL	;PUT CR AT END
CCE2 36 80	0381	LB (HL),CR	
CCE4 21 14 8E	0382	LB HL,BUFFER	;POINT TO BUFFER
CCE7 CB 4B CE	0383	CALL OUTPUT	;PRINT LINE
CCEA C1	0384	POP BC	;GET COUNT BACK
CCEB 79	0385	LB A,C	;WAS IT ALL?
CCEC A7	0386	AND A	;CHECK
CCE0 28 CD	0387	JR Z,XREF1	;YES,TRY NEXT
CCEF 21 14 8E	0388	LB HL,BUFFER	;CLEAR START OF BUFFER
CCF2 86 83	0389	LD B,5	
CCF4 36 2B	0390	LD (HL),'	
CCF6 23	0391	INC HL	
CCF7 1B FB	0392	BINZ ++5	
CCF9 EB	0393	EX DE,HL	;ADDRESS TO DE
CCFA 86 88	0394	LD B,8	;RESET B
CCFC 1B CC	0395	JR XREF2	;DO NEXT LINE
CCFE	0396	*****	
CCFE	0397	UTILITY SUBROUTINE - GETS TWO ADDRESSES	
CCFE	0398	TO BC & DE. CHECKS BC>=DE	
CCFE	0399	*****	
CCFE DF	0400 GETTWO:	RST SCAL	;READ LINE
CCFF 63	0401	DEFB INLINE	
CD09 JE 1B	0402	LB A,ESC	;ENSURE LINE CLEAR
CB02 F7	0403	RST ROUT	
CB03 B5	0404	PUSH DE	;SAVE POINTER
CB04 DF	0405	RST SCAL	;GET ADDRESSES FROM IT
CB05 79	0406	DEFB RLIN	
CB06 B1	0407	POP DE	;RESET DE
CB07 1A	0408	LB A,(DE)	;LOAD FIRST CHARAC. IN LINE
CB08 B8	0409	RET C	;RETURN ON ERROR
CB09 3A 8B 9C	0410	LB A,(ARGN)	;GET # ENTERED
CB0C FE 82	0411	CP 2	;TWO INT?
CB0E 37	0412	SCF	;SET ERROR FLAG

Interactive control program for REVAS

```

CB8F C8      9413    RET   NZ      ;RETURN IF NOT TWO
CB19 DF      9414    RST   SCAL    ;OK,GET THEM
CB11 68      9415    BEFB  ARGS
CB12 44      9416    LD    B,H    ;HL->BC
CB13 4B      9417    LD    C,L
CB14 B7      9418    OR    A      ;CLEAR CARRY
CB15 28      9419    BEC   HL
CB16 EB 52      9420    SBC   HL,DE    ;CHECK 2ND>1ST
CB18 3F      9421    CCF
CB19 C9      9422    RET
CB1A          9423    =====
CB1A          9424    ; ROUTINES TO PERFORM A COMPLETE PASS
CB1A          9425    =====
CB1A          9426    ;
CB1A          9427    ; SUBROUTINE TO GET NEXT SET OF ADDRESSES
CB1A          9428    ; FOR PASS
CB1A          9429    ;
CB1A 2A 4C 8E 9430    NEXTAD: LD    HL,(POINT)  ;GET ADDRESS POINTER
CB1D 28      9431    DEC   HL
CB1E 56      9432    LB    D,(HL)    ;PICK UP START ADDRESS
CB1F 28      9433    DEC   HL
CB20 5E      9434    LB    E,(HL)
CB21 22 4C 8E 9435    LD    (POINT),HL  ;PUT POINTER BACK
CB24 E5      9436    PUSH  HL    ;HL->IX
CB25 8D E1      9437    POP   IX
CB27 62      9438    LB    H,D    ;HL=DE
CB28 6B      9439    LB    L,E
CB29 ED 4B 34 8E 9440    LB    BC,(OFFSET) ;ABBR. OFFSET
CB2D 97      9441    ADD   HL,BC    ;COMPUTE PC
CB2E 44      9442    LB    B,H
CB2F 40      9443    LB    C,L
CB30 8D 46 FF 9444    LD    H,(IX-1)  ;GET END ADDRESS
CB33 8D 5E FE 9445    LD    L,(IX-2)
CB36 C9      9446    RET
CB37          9447    ;
CB37          9448    ; DO COMPLETE PASS
CB37          9449    ;
CB37 22 81 8E 9450    PASS:   LD    (PATCH+1),HL ;SET O/P DRIVER ADDR.
CB3A 2A 58 8E 9451    LD    HL,(SYNEND) ;INIT. SYMBOL TABLE POINTER
CB3B 22 52 8E 9452    LD    (PSYN),HL
CB49 2A 46 8E 9453    LD    HL,(START) ;INITIALISE POINTER ADDRESS..
CB43 22 4C 8E 9454    LD    (POINT),HL ;...TO MEMORY SECTIONS
CB46 CD 1A CD 9455    PASSL:  CALL   NEXTAD ;GET PAIR OF ADDRESSES
CB49 #3      9456    INC   BC    ;CORRECT THEM
CB4A 13      9457    INC   DE
CB4B 28      9458    DEC   HL
CB4C CD 3E C4 9459    CALL   REVAS   ;DO THEM
CB4F CD 1A CD 9460    CALL   NEXTAD ;GET DATA PAIR
CB52 78      9461    LB    A,L    ;CHECK FOR END OF TABLE
CB53 A4      9462    AND   H
CB54 3C      9463    INC   A
CB55 C8      9464    RET   Z      ;RETURN IF HL=$FFFF
CB56 3A 44 8E 9465    LD    A,(FLAG) ;CHECK PASS#
CB59 E6 89      9466    AND   $80    ;IF PASS 1, DON'T BOTHER...
CB5B 28 E9      9467    JR    Z,PASSL ;...WITH DATA AREAS.
CB5D          9468    ;
CB5D          9469    ; SECTION TO DEAL WITH DATA AREAS
CB5D          9470    ; PRINT AS:-
CB5D          9471    ; AAAA XX XX XX (LABEL) DB $XX,$XX,$XX ;C
CB5D          9472    ;

```

Interactive control program for REVAS

CD5D ED 5B 96 BE	9473	LD DE,(HADDR)	;GET NEXT ADDRESS
CD61 ED 52	9474	SBC HL,DE	;COMPUTE BYTE COUNT
CD63 38 E1	9475	JR C,PASSL	;LOOP IF -VE
CD65 23	9476	INC HL	;CORRECT IT
CD66 E5	9477	PUSH HL	;SAVE COUNT
CD67 CD 6E C4	9478 DATA1:	CALL INITB	;INITIALISE BUFFER
CD6A 94 93	9479	LD B,3	;3 BYTES/LINE
CD6C CD 88 C4	9480 DATA2:	CALL BYTE	;GET NEXT BYTE
CD6F E1	9481	POP HL	;RECOVER COUNT
CD70 2B	9482	DEC HL	;MORE?
CD71 7D	9483	LD A,L	
CD72 B4	9484	OR H	
CD73 E5	9485	PUSH HL	;SAVE COUNT
CD74 28 92	9486	JR Z,*+2	;SKIP CHECK ON B IF DONE
CD76 18 F4	9487	DJNZ DATA2	;MORE ON LINE?
CD78 F5	9488	PUSH AF	;SAVE FLAG
CD79 CD 53 C9	9489	CALL NOTVAL	;PUT IN 'DEFB' BIT
CD7C 21 14 9E	9490	LD HL,BUFFER	;PRINT
CD7F CD 25 CE	9491	CALL PASS2	
CD82 F1	9492	POP AF	;RECOVER FLAG
CD83 29 E2	9493	JR NZ,DATA1	;LOOP IF MORE
CD85 E1	9494	POP HL	;REMOVE COUNT
CD86 18 BE	9495	JR PASSL	;ELSE NEXT SECTION
CD88	9496 ;*****		
CD88	9497 ; PASS1 OUTPUT SECTION - BUILDS SYMBOL TABLE		
CD88	9498 ;*****		
CD88 2A 12 BE	9499 PASS1:	LD HL,(LABELP)	;CHECK IF ANY REF.
CD8B 7D	9500	LD A,L	
CD8C B4	9501	OR H	
CD8D C8	9502	RET Z	;RETURN IF NONE
CD8E 2A 58 BE	9503	LD HL,(SYMEND)	;LOAD TABLE ADDRESS
CD91	9504 ;		
CD91	9505 ; NOW FIND SYMBOL IN TABLE		
CD91	9506 ; TABLE IS ORGANISED AS:-		
CD91	9507 ; <NENTRIES> <LABEL><ADDR.> <REF><ADDR1> <REF><ADDR2> ...		
CD91	9508 ; "LABELS" ARE IN ENTERED IN ASCENDING NUMERICAL ORDER		
CD91	9509 ; BECAUSE OF ZEP FILE THE TABLE STARTS AT THE		
CD91	9510 ; TOP OF MEMORY AND WORKS IT WAY DOWN.		
CD91	9511 ; EOT MARKED BY \$FF		
CD91	9512 ;		
CD91 7E	9513 SRCHL:	LD A,(HL)	;LOAD # ENTRIES
CD92 E5	9514	PUSH HL	;GET HL TO IT
CD93 FD E1	9515	POP IY	
CD93 3C	9516	INC A	;TEST
CD94 28 2E	9517	JR Z,PAST	;EDIT
CD95 47	9518	LD B,A	;COUNT IN B
CD97 2B	9519	DEC HL	
CD9A 5E	9520	LD E,(HL)	;LOAD LABEL IN TABLE
CD9B 2B	9521	DEC HL	
CD9C 36	9522	LD D,(HL)	
CD9D 2B	9523	DEC HL	
CD9E E5	9524	PUSH HL	;SAVE POINTER
CD9F 2A 18 BE	9525	LD HL,(LABELA)	;GET CURRENT LABEL
CDA2 A7	9526	AND A	;CLEAR CARRY
CDA3 ED 52	9527	SBC HL,DE	;COMPARE
CDA3 E1	9528	POP HL	;RECOVER POINTER
CDA6 18 92	9529	JR *+2	;SKIP IN CASE NONE
CDA8 2B	9530	DEC HL	;SPACE TO END OF ENTRY
CDA9 2B	9531	DEC HL	
CDAA 18 FC	9532	DJNZ *-4	

Interactive control program for REVAS

```

CDAC 3A 44 9E      #533 LD A,(FLAG) ;GET FLAG TO A
CDAF 38 15      #534 JR C,PAST ;NEW < TABLE ENTRY
CDB1 29 DE      #535 JR NZ,SRCHL ;TRY NEXT ENTRY
CDB3          #536 ;
CDB3          #537 ; FOUND ENTRY, SO APPEND XREF TO IT
CDB3          #538 ; ONLY IF REQUIRED.
CDB3          #539 ;
CDB3 CB 5F      #540 BIT 3,A ;CHECK IF "X" OPTION.
CDB3 C8      #541 RET Z ;RETURN IF NOT
CDB6 9E 92      #542 LD C,2 ;MAKE ROOM FOR IT
CDB8 CD EF CD      #543 CALL MVUP
CDB8 EB 5B 9E 9E      #544 FND: LD DE,(LINEA) ;INSERT XREF
CDBF 73      #545 LD (HL),E
CDC# 28      #546 DEC HL
CDC1 72      #547 LD (HL),D
CDC2 FD 34 00      #548 INC (IY) ;BUMP COUNT
CDC3 C9      #549 RET ;DONE
CDC6          #550 ;
CDC6          #551 ; NEW ENTRY WANTED, MAKE
CDC6          #552 ; ROOM AND INSERT IT
CDC6          #553 ; BUT FIRST UPDATE LABEL COUNT IN CASE ZEAP FILE
CDC6          #554 ;
CDC6 2A 82 9E      #555 PAST: LB HL,(ZLABEL)
CDC9 23      #556 INC HL
CDCA 23      #557 INC HL
CDCB 22 82 9E      #558 LD (ZLABEL),HL
CDCE FD E3      #559 PUSH IY ;BACK TO START OF ENTRY
CDC9 E1      #560 POP HL
CDC1 3A 44 9E      #561 LD A,(FLAG) ;GET FLAG BYTE
CDC4 9E 95      #562 LD C,3 ;MAKE ROOM FOR NEW ENTRY
CDC6 CB 5F      #563 BIT 3,A ;XREF?
CDC8 29 92      #564 JR NZ,*+2 ;YES SKIP
CDC8 9E 93      #565 LD C,3 ;NO,SO NO XREF ENTRY
CDCD CD EF CD      #566 CALL MVUP
CDCF 34 00      #567 LD (HL),9 ;COUNT=9
CDE1 ED 5B 19 9E      #568 LD DE,(LABELA) ;PUT LABEL IN
CDE5 28      #569 DEC HL
CDE6 73      #570 LD (HL),E
CDE7 28      #571 DEC HL
CDE8 72      #572 LD (HL),D
CDE9 28      #573 DEC HL
CDEA CB 5F      #574 BIT 3,A ;"XREF" GOING?
CDEC C8      #575 RET Z ;NO,DONE
CDED 18 CC      #576 JR FND ;REST IN 'FOUND'
CDEF          #577 ;
CDEF          #578 ; MAKE SPACE WITHIN THE TABLE
CDEF          #579 ; FOR A NEW ENTRY & CHECK FOR
CDEF          #580 ; OVERFLOW OF THE TABLE
CDEF          #581 ;
CDEF ED 5B 52 9E      #582 MVUP: LD DE,(PSYM) ;SEE HOW MUCH TO MOVE
CDF3 23      #583 INC HL ;CLEAR CARRY
CDF4 A7      #584 AND A ;CLEAR B
CDF5 96 98      #585 LD B,B ;HL-BYTE COUNT
CDF7 ED 52      #586 SBC HL,DE ;COUNT TO DE
CDF9 EB      #587 EX DE,HL ;HL=NEW END
CDFA ED 42      #588 SBC HL,BC ;BC=NEW END
CDFC 42      #589 LD B,D ;COUNT TO BC
CDFD 4B      #590 LD C,E
CDFE EB      #591 EX DE,HL ;NEW END TO DE
CDFF 2A 4E 9E      #592 LD HL,(SYNST) ;CHECK FOR TABLE OVERFLOW

```

Interactive control program for REVAS

CE92 2B	0593	BEC	HL	
CE93 ED 52	0594	SBC	HL,DE	
CE95 38 98	0595	JR	NC,DVRFNU	
CE97 2A 32 9E	0596	LD	HL,(PSYM)	;RESET SOURCE ADDRESS
CE9A ED 53 52 9E	0597	LD	(PSYM),DE	;UPDATE PSYM
CE9E ED 30	0598	LDIR		;MOVE TABLE DOWN
CE10 2B	0599	BEC	HL	;ADJUST ADDRESS
CE11 C9	0600	RET		
CE12	0601 ;			
CE12 EF	0602 DVRFNU:	RST	PRS	;ERROR MESSAGE
CE13 4F 76 65 72	0603	BEFB	'Overflow',CR,#	
66 6C 6F 77				
6D 99				
CE18 21 14 9E	0604	LD	HL,BUFFER	;PRINT CURRENT LINE
CE20 CD F8 CE	0605	CALL	OUTLIN	
CE23 DF	0606	RST	SCAL	
CE24 58	0607	DEFB	RETURN	
CE25	0608 ;=====			
CE25	0609 ; PASS 2 - PRINTS OUTPUT & ADDS LABELS			
CE25	0610 ;=====			
CE25 #1 13 98	0611 PASS2:	LD	BC,19	;SPACE PAST OPCODES ETC
CE28 FD 21 58 9E	0612	LD	IY,SEQU	;IN CASE EQU WANTED
CE2C E3	0613	PUSH	HL	;GET HL TO IX
CE2D DD E1	0614	POP	IX	
CE2F DD 99	0615	ADD	IX,BC	;IX POINTS TO LABEL POSITION
CE31 3A 44 9E	0616	LD	A,(FLAG)	
CE34 CB 47	0617	BIT	8,A	;OPCODES ETC WANTED?
CE36 28 #3	0618	JR	Z,++3	;YES
CE38 99	0619	ADD	HL,BC	;NO,ALTER BUFFER START ADDR.
CE39 FD 99	0620	ADD	IY,BC	
CE3B CB 67	0621	BIT	4,A	;LABELS?
CE3D 28 9E	0622	JR	Z,OUTPUT	;NONE,SO SKIP
CE3F E3	0623	PUSH	HL	;SAVE ADDRESS
CE40 CD 92 CF	0624	CALL	LABEL	;DO LABEL CHECK
CE43 2A 12 9E	0625	LD	HL,(LABELP)	;CHECK OPERAND FOR...
CE46 7B	0626	LD	A,L	;..LABEL THERE
CE47 34	0627	OR	H	
CE48 28 #2	0628	JR	Z,++2	;SKIP IF NONE
CE4A 36 4C	0629	LD	(HL),'L'	;CHANGE 'Y' TO 'L'
CE4C E1	0630	POP	HL	;RECOVER ADDRESS
CE4D	0631 ;			
CE4D	0632 ; NOW OUTPUT THE LINE			
CE4D	0633 ;			
CE4D 44	0634 OUTPUT:	LD	B,H	;TEMP SAVE IN BC
CE4E 48	0635	LD	C,L	
CE4F 2A 4A 9E	0636	LD	HL,(LAST)	;CHECK AGAINST RANGE
CE52 ED 58 9E 9E	0637	LD	DE,(LINEA)	
CE56 A7	0638	AND	A	;CLEAR CARRY
CE57 ED 52	0639	SBC	HL,DE	
CE59 DB	0640	RET	C	;RETURN IF LINEA>LAST
CE5A 2A 48 9E	0641	LD	HL,(FIRST)	
CE5D EB	0642	EX	DE,HL	
CE5E EB 52	0643	SBC	HL,DE	
CE6B DB	0644	RET	C	;RETURN IF FIRST<LINEA
CE61 C9	0645	PUSH	BC	;SAVE POINTER ON STACK
CE62	0646 ;			
CE62 JA 44 9E	0647	LD	A,(FLAG)	;CHECK OPTIONS
CE65 F5	0648	PUSH	AF	;SAVE ON STACK
CE66 CB 6F	0649	BIT	3,A	;PAGED?
CE68 C4 7B CF	0650	CALL	nz,BPAGE	;YES, DEAL WITH IT

Interactive control program for REVAS

```

CE6B F1          8631    POP AF      ;(RESET FLAG)
CE6C E1          8632    POP HL      ;RECOVER POINTER
CE6D F5          8633    PUSH AF     ;SAVE FLAG AGAIN
CE6E 9F          8634    RRCA
CE6F 30 3D       8635    JR NC,ALLIN  ;SKIP IF ALL WANTED
CE71             8636 ;*****CRUNCH SOURCE INTO BUFFER (USE EQU ONE)*****
CE71             8637 ; CRUNCH SOURCE INTO BUFFER (USE EQU ONE)
CE71             8638 ;*****START OF BUFFER*****
CE71 EB          8639    EX DE,HL    ;TEMP SAVE HL
CE72 21 99 99     8660    LD HL,9     ;OFFSET TO 4TH BYTE OF LINE.
CE73 19          8661    ADD HL,DE   ;POINT TO IT
CE76 7E          8662    LD A,(HL)   ;GET IT
CE77 EB          8663    EX DE,HL    ;RESET HL
CE78 11 38 9E     8664    LD DE,SEQU  ;START OF BUFFER
CE78 91 38 28     8665    LD BC,' '  ;SET FOR NOT 'DEFB'
CE7E 96 42          8666    SUB 'B'    ;DEFB?
CE80 29 91          8667    JR NZ,++1  ;NO,SKIP
CE82 4F          8668    LD C,A     ;YES,CLEAR DELIMITER
CE83 7E          8669 SQSH1: LD A,(HL)   ;LOAD BYTE
CE84 FE 90          8670    CP CR      ;REACHED END?
CE86 28 19          8671    JR Z,SQSH3  ;YES
CE88 B9          8672    CP C        ;REACHED DELIMITER?
CE89 28 16          8673    JR Z,SQSH3  ;YES,JUMP ON
CE8B 12          8674    LD (DE),A   ;NO,COPY IT OVER
CE8C 13          8675    INC DE      ;UPDATE ADDRESSES
CE8D 23          8676    INC HL      ;WAS IT DEFN DELIMITER?
CE8E FE 2F          8677    CP '/'     ;NO,SKIP
CE8F 29 96          8678    JR NZ,SQSH2  ;YES,MODIFY OR RESET
CE92 78          8679    LD A,B      ;NO,MODIFY OR RESET
CE93 2F          8680    CPL
CE94 47          8681    LD B,A     ;NO,CONTINUE
CE95 79          8682    LD A,C
CE96 2F          8683    CPL
CE97 4F          8684    LD C,A
CE98 B8          8685 SQSH2: CP B        ;SPACE?
CE99 28 E8          8686    JR NZ,SQSH1  ;NO,CONTINUE
CE9B 9E          8687    CP (HL)    ;NEXT A SPACE?
CE9C 28 E5          8688    JR NZ,SQSH1  ;NO,COPY IT OVER(?)
CE9E 23          8689    INC HL      ;YES,IGNORE IT
CE9F 18 F7          8690    JR SQSH2   ;GO CHECK NEXT
CEA1             8691 ; DELETE ANY ODD SPACE AT THE END OF THE LINE
CEA1 EB          8692 SQSH3: EX DE,HL    ;DE->HL
CEA2 28          8693    DEC HL
CEA3 7E          8694    LD A,(HL)   ;GET LAST BYTE
CEA4 FE 29          8695    CP '/'     ;SPACE?
CEA6 28 91          8696    JR Z,++1  ;YES,SKIP
CEA8 23          8697    INC HL      ;NO SET CR AT END
CEA9 36 90          8698    LD (HL),CR
CEAB 21 38 9E     8699    LD HL,SEQU  ;RESET BUFFER POINTER
CEAE             8700 ;*****PRINT THE LINE AND ALSO MOVE TO ZEAP FILE (?)*****
CEAE             8701 ; PRINT THE LINE AND ALSO MOVE TO ZEAP FILE (?)
CEAE             8702 ;*****PUT LINE NO INTO FILE*****
CEAE CD FB CE     8703 ALLIN: CALL OUTLIN
CEB1 F1          8704    POP AF      ;RECOVER FLAG
CEB2 CB 57          8705    BIT 2,A     ;CHECK FOR ZEAP FILE
CEB4 C8          8706    RET Z      ;NONE,SO RETURN
CEB5 ED 3B 99 9E     8707    LD DE,(LINENO) ;GET CURRENT LINE NUMBER
CEB9 2A 56 9E     8708    LD HL,(ZEAPP) ;AND POINTER INTO ZEAP FILE
CEBC 23          8709    INC HL      ;SPACE PAST NULL
CEBD 73          8710    LD (HL),E   ;PUT LINE NO INTO FILE

```

Interactive control program for REVAS

CEBE 23	9711	INC	HL
CEBF 72	9712	LD	(HL),D
CEC0 23	9713	INC	HL
CEC1 EB	9714	EX	DE,HL ;FILE POINTER TO DE
CEC2 3E 81	9715	LD	A,1 ;UPDATE LINE NO.
CEC4 83	9716	ADD	L
CEC5 27	9717	DAA	
CEC6 6F	9718	LD	L,A
CEC7 3E 98	9719	LD	A,B
CEC9 8C	9720	ADC	H
CECA 27	9721	DAA	
CECB 67	9722	LD	H,A
CECC 22 68 8E	9723	LD	(LINENO),HL
CECF 21 58 8E	9724	LD	HL,SEQU ;SET SOURCE POINTER
CED2 3E 9D	9725	LD	A,CR ;MOVE LINE TO FILE
CED4 ED A8	9726 Z1:	LDI	
CED6 BE	9727	CP	(HL) ;END YET?
CED7 28 FB	9728	JR	NZ,Z1 ;NO,LOOP
CED9 ; NOW SAVE POINTER AND UPDATE FILE START & END	9729		
CED9 EB	9730	EX	DE,HL ;POINTER BACK TO HL
CEBA 22 56 8E	9731	LD	(ZEAPP),HL ;UPDATE IN MEM.
CEDD 36 98	9732	LD	(HL),B
CEDF 23	9733	INC	HL
CEE9 36 FF	9734	LD	(HL),\$FF
CEE2 23	9735	INC	HL
CEE3 E3	9736	PUSH	HL ;SAVE FOR OVERFLOW CHECK
CEE4 ED 5B 99 8F	9737	LD	DE,(SF88) ;COMPUTE REL. OFFSET
CEE8 ED 52	9738	SBC	HL,DE
CEEA EB	9739	EX	DE,HL
CEE9 73	9740	LD	(HL),E ;SET AT FILE START
CEEC 23	9741	INC	HL
CEED 72	9742	LD	(HL),D
CEEE D1	9743	POP	DE ;RECOVER END POINTER
CEF1 2A 4E 8E	9744	LD	HL,(SYMST) ;CHECK FOR OVERFLOW
CEF2 ED 52	9745	SBC	HL,DE
CEF4 D9	9746	RET	NC ;OK IF BELOW SYMBOL TABLE
CEF5 C3 12 CE	9747	JP	OVRFLU
CEF8 ;*****	9748		
CEF8 ; OUTPUT A LINE TO THE SCREEN (AND TAPE?)	9749		
CEF8 ;*****	9750		
CEF8 46	9751 OUTLIN:	LD	B,(HL) ;GET BYTE
CEF9 CD AA CF	9752	CALL	CHROUT ;OUTPUT IT
CEF0 23	9753	INC	HL
CEF0 FE 9B	9754	CP	CR ;FINISHED YET?
CEFF 28 F7	9755	JR	NZ,OUTLIN ;LOOP IF MORE
CF91 C9	9756	RET	
CF92 ;*****	9757		
CF92	9758 ;		PASS 2 - LABEL HANDLING
CF92 ;*****	9759		
CF92 2A 52 8E	9760 LABEL:	LD	HL,(PSYM) ;POINT TO CURRENT SYMBOL
CF95 7E	9761	LD	A,(HL) ;LOAD # ENTRIES
CF96 3C	9762	INC	A ;CHECK FOR EOT
CF97 C8	9763	RET	Z ;RETURN IF THERE
CF98 47	9764	LD	B,A ;SAVE COUNT
CF99 2B	9765	DEC	HL ;GET SYMBOL
CF9A 3E	9766	LD	E,(HL) ;...TO DE
CF9B 2B	9767	DEC	HL
CF9C 54	9768	LD	D,(HL)
CF9D 2B	9769	DEC	HL
CF9E 18 82	9770	JR	**2 ;SKIP IN CASE NONE

Interactive control program for REVAS

CF18 2B	9771	DEC HL	;SPACE TO END OF ENTRY
CF11 2B	9772	DEC HL	
CF12 1B FC	9773	DJNZ *-4	
CF14 44	9774	LD B,H	;SAVE POINTER IN BC
CF15 4D	9775	LD C,L	
CF16 2A 9E 9E	9776	LD HL,(LINEA)	;GET CURRENT LINE
CF19 3A 44 9E	9777	LD A,(FLAG)	;CHECK IF FORCING REMAINING..
CF1C 97	9778	RLCA	;...EQU'S. PASS IN CARRY.
CF1D 3F	9779	CCF	;0=NORMAL PASS2,1=FORCE EQU'S
CF1E 38 93	9780	JR C,*+3	;1=SKIP CHECK
CF20 ED 52	9781	SDC HL,DE	;COMPARE WITH ENTRY
CF22 D8	9782	RET C	;RETURN,NOT THERE YET
CF23 ED 43 52 9E	9783	LD (PSYM),BC	;SET NEXT ENTRY ADDRESS
CF27 2B 4B	9784	JR Z,ALAB	;EQUAL,SO APPEND LABEL
CF29	9785 ;		
CF29	9786 ; PAST LABEL, SO INSERT 'EQU'		
CF29	9787 ;		
CF29 44	9788	LD B,H	;GET OFFSET TO BC
CF2A 4D	9789	LD C,L	
CF2B C5	9790	PUSH BC	;SAVE BC
CF2C 94 13	9791	LD B,19	
CF2E 21 58 9E	9792	LD HL,SEOU	;CLEAR BUFFER FOR IT
CF31 36 2B	9793	LD (HL),' '	
CF33 23	9794	INC HL	
CF34 1B FB	9795	DJNZ *-5	
CF36 F5	9796	PUSH AF	;PRESERVE A
CF37 CD 74 CF	9797	CALL ALAB	;WRITE LABEL
CF3A F1	9798	POP AF	;RESET IT
CF3B E5	9799	PUSH HL	;SAVE HL
CF3C 21 F7 CF	9800	LD HL,EQU	;PUT IN 'EQU'
CF3F 91 94 98	9801	LD BC,6	
CF42 ED 98	9802	LDIR	
CF44 EB	9803	EX DE,HL	
CF45 D1	9804	POP DE	;LABEL' TO DE
CF46 C1	9805	POP BC	;RESET BC
CF47 9F	9806	RRCA	;''FORCE'' FLG STILL IN A
CF48 3B 15	9807	JR NC,ABSOL	;''FORCE''=ABSOLUTE LABEL
CF4A 78	9808	LD A,B	;CHECK FOR RANGE
CF4B A7	9809	AND A	;ABSOLUTE IF >255
CF4C 2B 11	9810	JR NZ,ABSOL	
CF4E 79	9811	LD A,C	
CF4F FE 95	9812	CP 3	;ABSOLUTE IF >4
CF51 3B 9C	9813	JR NC,ABSOL	
CF53 36 24	9814	LD (HL),'\$'	;INSERT PC. RELATIVE
CF55 23	9815	INC HL	
CF56 36 2D	9816	LD (HL),'~'	
CF58 23	9817	INC HL	
CF59 F6 38	9818	OR \$38	;MAKE OFFSET ASCII
CF5B 77	9819	LD (HL),A	;PUT IN BUFFER
CF5C 23	9820	INC HL	;ADD A CR
CF5B 1B 98	9821	JR PEQU	;PRINT LINE
CF5F	9822 ;		
CFSF	9823 ; ABSOLUTE LABEL		
CFSF	9824 ;		
CF5F 36 23	9825 ABSOL:	LD (HL),'#'	;HEX
CF61 23	9826 INC HL		;INSERT ADDRESS
CF62 EB	9827 EX DE,HL		
CF63 CD 9D C4	9828 CALL HEX4		
CF66 EB	9829 EX DE,HL		
CF67 36 9D	9830 PEQU:	LD (HL),CR	;SET CR AT END

Interactive control program for REVAS

CF69 FD E3	9831	PUSH IX	;GET IX TO HL
CF6B E1	9832	POP HL	
CF6C CB 4D CE	9833	CALL OUTPUT	;PRINT LINE
CF6F 18 91	9834	JR LABEL	;CHECK NEXT LABEL
CF71	9835 ;		
CF71	9836 ; WRITE LABEL		
CF71	9837 ;		
CF71 BB E3	9838 ALAB:,	PUSH IX	;GET IX TO HL
CF73 E1	9839	POP HL	
CF74 36 4C	9840 ALAB1:	LD (HL), 'L'	;LEADING L
CF76 23	9841	INC HL	
CF77 E3	9842	EX DE,HL	;WRITE TO BUFFER
CF78 C3 98 C4	9843	JP HEX4	;RETURN THRU' HEX4
CF78	9844 ;*****		
CF78	9845 ; PAGED,CHECK LINE COUNT ETC		
CF78	9846 ;*****		
CF78 21 7C 9E	9847 DPAGE:	LD HL,LC	;POINT TO LC
CF7E 35	9848	BEC (HL)	
CF7F C8	9849	RET NZ	;RETURN IF NOT EOP
CF80 3A 70 9E	9850 EJECT:	LD A,(GAP)	;COMPUTE # LINE FEEDS
CF83 86	9851	ADD (HL)	
CF84 57	9852	LD B,A	;SAVE IN B
CF85 96 98	9853	LD B,CR	
CF87 18 93	9854	JR ++3	;SKIP
CF89 CD AA CF	9855	CALL CHROUT	;DO REQUISITE #
CF8C 15	9856	DEC B	
CF8D 28 FA	9857	JR NZ,++6	
CF8F 2A 7E 9E	9858	LD HL,(LINES)	;RESET LINE COUNT
CF92 22 7C 9E	9859	LD (LC),HL	;..(AND GAP)
CF95 21 80 9E	9860	LD HL,PAGE+1	;INC PAGE NUMBER
CF98 34	9861	INC (HL)	
CF99 7E	9862	LD A,(HL)	;CHECK FOR CARRY
CF9A FE 3A	9863	CP '9'+1	
CF9C 28 84	9864	JR NZ,NOINC	
CF9E 36 38	9865	LD (HL),'9'	;RESET
CFA0 28	9866	DEC HL	
CFA1 34	9867	INC (HL)	;BUMP PAGE NUMBER
CFA2 21 84 9E	9868 NOINC:	LD HL,HEAD	;PRINT HEADING
CFA5 CB F8 CE	9869	CALL OUTLIN	
CFA8 96 9D	9870	LD B,CR	;SPACE ONE LINE
CFAA	9871 ;*****		
CFAA	9872 ; CHARACTER OUTPUT ROUTINE		
CFAA	9873 ;*****		
CFAA 78	9874 CHROUT: LD A,B		;O/P TO SCREEN
CFAB F7	9875	RST ROUT	
CFAC E5	9876	PUSH HL	;SAVE HL
CFAD 21 44 9E	9877	LD HL,FLAG	;POINT TO FLAG
CFB0 CD BF CF	9878	CALL PUNCH	;PUNCH IT MAYBE
CFB3 FE 9D	9879	CP CR	;EDLT?
CFB5 28 26	9880	JR NZ,EXCHR	;NO, EXIT
CFB7 3E 9A	9881	LD A,LF	;ADD LF MAYBE
CFB9 CB BF CF	9882	CALL PUNCH	
CFBC 3A 45 9E	9883	LD A,(STEP)	;CHECK SINGLE STEP FLAG
CFBF FE 20	9884	CP //	;SPACE?
CFC1 28 84	9885	JR Z,WAIT	;IF S/STEP SKIP NEXT
CFC3 DF	9886	RST SCAL	;POLL THE KEYBOARD
CFC4 62	9887	BEFB POLLX	
CFC5 39 84	9888	JR NC,++4	;IF NOTHING-CNTUE
CFC7 CF	9889 WAIT:	RST RIN	;READ A CHARACTER
CFC8 32 45 9E	9890	LD (STEP),A	;SAVE KEY

Interactive control program for REVAS

CFCB FE 1B	0891	CP ESC	;ESCAPE PRESSED?
CFCD CA A2 CC	0892	JP Z,EXIT	;YES,RETURN
CFD# 78	0893	LD A,B	;RESET A)
CFD1 CB 76	0894	BIT 6,(HL)	;DELAY GOING?
CFD3 2B 98	0895	JR Z,EXCHR	;NO,EXIT
CFD5	0896 ; SHORT DELAY ROUTINE		
CFD5 C5	0897	PUSH BC	;PRESERVE BC -(FIRST LOOP SHORT)
CFD6 #6 89	0898	LD B,128	;LOOP COUNT
CFD8 FF	0899 DLOOP:	RST RDEL	;DELAY
CFD9 18 FD	0900	DJNZ DLOOP	;DELAY
CFD8 C1	0901	POP BC	;RESTORE BC
CFDC 78	0902	LD A,B	;AND A
CFDD E1	0903 EXCHR:	POP HL	;RESET HL
CFDE C9	0904	RET	
CFDF	0905 ;		
CFDF CB 4E	0906 PUNCH:	BIT 1,(HL)	;TAPE OPTION ON?
CFE1 C8	0907	RET Z	;NO,RETURN
CFE2 DF	0908	RST SCAL	;PUNCH IT
CFE3 6F	0909	DEFB SRlx	;D/P TO SERIAL PORT
CFE4 C9	0910	RET	
CFE5	0911 ;*****		
CFE5	0912 ; RAM INITIALISATION IMAGE		
CFE5	0913 ;*****		
CFE5 1B 88 98 99	0914 RAMLD:	DEFW \$1B,8	
CFE9 2B 2B 2B 58	0915	DEFB ' PAGE 89	
41 47 45 28			
38 39 2B 29			
2B 2B			
8012	0916 MOVLNG:	EQU *-RAMLD	
CFE7 2B 45 51 55	0917 EQU:	DEFB ' EQU '	
2B 2B			
CFD0 C3 3E C4	0918	JP REVAS	;Fixed transfer to REVAS
	0929	END	

CROSS REFERENCE

ABSOL	CF5F	9097 9819 9813
ADDR	9E84	
ALAB	CF71	9784
ALABI	CF74	9797
ALLINS	CC8C	9313
ALLLIN	CEAE	9455
ARGH	9C9B	9199 9419
ARGS	9868	9293 9415
ASKT	CB4E	9171
BLINK	997B	9299
BUFFER	9E14	9295 9269 9289 9367 9382 9388 9496 9494
BYTE	C488	9489
CHROUT	CFAA	9752 9855
CR	986D	9898 9111 9147 9161 9189 9193 9227 9236 9381 9603 9679 9498 9725 9754 9838 9853 9870 9879
CRLF	996A	9392
DA	C8DC	9224
DAREA	CBE4	9274
DAREA8	CBF4	9267 9281 9286
DAREA1	CC29	9248
DAREA2	CC37	9259 9242 9271
DAREA3	CC49	9269
DATA1	CD67	9493
DATA2	CD6C	9487
DLOOP	CFD8	9986
DPAGE	CF7B	9858
EADDR	9E88	
EJECT	CF89	9353
EDU	CF77	9809
ERRH	986B	9283
ESC	9818	9286 9288 9402 9891
EXCHR	CFDD	9880 9895
EXIT	CCA2	9363 9892
FIRST	9E48	9238 9641
FLAG	9E44	9117 9222 9396 9319 9336 9465 9533 9361 9616 9647 9777 9877
FND	CDBB	9576
FOUND	CAE4	9189
GAP	9E7B	9358 9859
GETSTA	CB11	9149
GETTVO	CCFE	9148 9228 9239
HEAD	9E84	9868
HEX4	C49D	9372 9828 9843
HXYFLG	9E8C	
INITB	C46E	9478
INLINE	9963	9163 9195 9491
LABEL	CF92	9349 9624 9834
LABELA	9E14	9325 9568
LABELP	9E12	9499 9625
LAST	9E4A	9231 9359 9636
LC	9E7C	9175 9847 9859
LF	988A	9881
LINEA	9E8E	9544 9637 9776
LINENG	9E89	9882 9797 9723
LINES	9E7E	9173 9858
LUOP	CAF6	9119
MADDR	9E96	9473
MOVING	9912	9884
MVUP	CDEF	9543 9366
NEXTAD	CD1A	9455 9469

CROSS REFERENCE

NLABS	CB49	\$133
NOINC	CFA2	\$864
NOTITL	CB86	\$159 \$198 \$291
NOTVAL	C953	\$489
MUM	\$864	\$178
MONY	\$C21	\$172
OFFSET	BE54	\$216 \$243 \$445
OPCAADR	BE6A	
OPTION	CAC6	\$112
OTAB	CAD2	\$186
OUTLIN	CEF8	\$693 \$793 \$755 \$869
OUTPUT	CE4D	\$383 \$622 \$833
OVRFLW	CE12	\$595 \$747
PAGE	SE8C	\$869
PASS	CD37	\$389 \$335
PASS1	CB88	\$388
PASS2	CE25	\$334 \$491
PASSL	CB46	\$462 \$475 \$495
PAST	CDC6	\$517 \$534
PATCH	SE98	\$458
PEQU	CF67	\$821
POINT	SE4C	\$438 \$435 \$454
PULLK	\$862	\$887
PRS	\$928	\$689 \$141 \$144 \$146 \$169 \$192 \$226 \$235 \$279 \$284 \$296 \$682
PSYN	SE52	\$322 \$452 \$582 \$596 \$597 \$760 \$783
PUNCH	CFDF	\$878 \$882
RANLD	CFE3	\$883 \$916
RDEL	\$838	\$899
RETURN	\$858	\$345 \$607
REVAS	C43E	\$459 \$918
REVASC	CABB	
RIN	\$888	\$182 \$889
RLIN	\$879	\$197 \$496
RNGRQ	CBC6	\$229
ROUT	\$938	\$193 \$493 \$875
RSTFLG	SE93	\$898
SAVE	CABB	\$878
SCAL	\$818	\$162 \$194 \$196 \$292 \$282 \$298 \$301 \$344 \$480 \$495 \$414 \$696 \$886 \$998
SEQU	SE58	\$612 \$664 \$699 \$724 \$792
SETLPP	CB69	\$167
SOSH1	CE83	\$686 \$688
SOSH2	CE98	\$678 \$699
SOSH3	CEA1	\$671 \$673
SRCHL	CD91	\$535
SRLX	\$86F	\$989
ST	CB22	\$149
ST#	CB2E	\$143
START	SE46	\$299 \$453
STEP	SE43	\$389 \$883 \$898
SYMEND	SE38	\$151 \$318 \$369 \$451 \$593
SYMST	SE4E	\$158 \$328 \$323 \$592 \$744
TITLE	SE92	\$177
WAIT	CFC7	\$885
XREF1	CCBC	\$387
XREF2	CCCA	\$378 \$395
ZI	CED4	\$728
ZEAPP	SE36	\$332 \$708 \$731
ZLABEL	SE82	\$324 \$555 \$558